NAVAL POSTGRADUATE SCHOOL
Monterey, California

(2)

THESIS

Optimizing Ship Air-Defense Evaluation Model
Using Simulation and Inductive Learning

by

Chang-yun Lo

March, 1991

Thesis Advisor: Yuh-jeng Lee

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) Naval Postgraduate School
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School — 6b. OFFICE SYMBOL (if applicable) OR	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000	7b. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION — 8b. OFFICE SYMBOL (if applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
OPTIMIZING SHIP AIR-DEFENSE EVALUATION MODEL USING SIMULATION AND INDUCTIVE LEARNING

12. PERSONAL AUTHOR(S) Lo, Chang-yun

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Master's Thesis		March, 1991	83

16. SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Ship Air-Defense Simulation Model, Inductive Learning, ID3

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This thesis presents an effective method to integrate simulation modeling with inductive learning to analyze ship air-defense combat scenarios. By combining the use of inductive learning with simulation, we are able to discover rules in a ship air-defense evaluation model about the optimal weapon assignments that we might not be aware of or could not express clearly. This approach can also perform sensitivity analysis in identifying variables that are critical for certain weapon operations. In addition, results obtained from inductive learning, as represented in the format of decision trees, are easy for a human user to understand, maintain, and adopt for other use.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL Yuh-jeng Lee	22b. TELEPHONE (Include Area Code) (408) 646-2361	22c. OFFICE SYMBOL CS/LE

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

Approved for public release; distribution is unlimited

Optimizing Ship Air-Defense Evaluation Model
Using Simulation and Inductive Learning

by

Lo, Chang-yun
Commander, Republic of China Navy
B.S., Chinese Naval Academy, 1978

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March, 1991

Author: *Lo, Chang-yun*

Approved By: Lee, Yuh-jeng, Thesis Advisor

Lyn R. Whitaker, Second Reader

P. Purdue, Chairman,
Department of Operations Research

ABSTRACT

This thesis presents an effective method to integrate simulation modeling with inductive learning to analyze ship air-defense combat scenarios. By combining the use of inductive learning with simulation, we are able to discover rules in a ship air-defense evaluation model about the optimal weapon assignments that we might not be aware of or could not express clearly. This approach can also perform sensitivity analysis in identifying variables that are critical for certain weapon operations. In addition, results obtained from inductive learning, as represented in the format of decision trees, are easy for a human user to understand, maintain, and adopt for other use.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. *Background*

Predicting combat effectiveness is a critical but difficult task. It is usually performed using testing data in analytical procedures involving a model, simulation, or game (i.e., the tools of operations research). More often than not, the evaluation of combat effectiveness cannot be deduced directly from test measures.

This thesis attempts to combine the use of inductive rule generation with a deductive simulation model to find the most effective way to assign weapons for ship air-defense. Figure 1-1 shows the conceptual model of our approach:

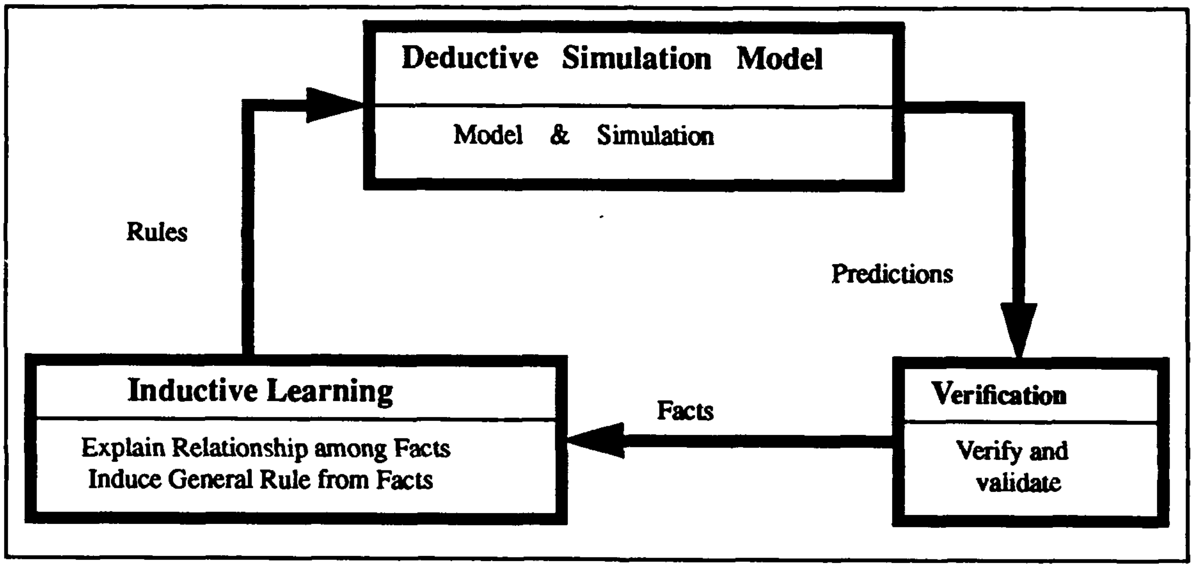

Figure 1-1 A Conceptual Model

B. *Deductive Simulation Model*

The scenario for this simulation model is a ship patrolling in an assigned region and performing area defense to destroy or neutralize threats from the air which can be either fighters or missiles. The ship is equipped with Anti-Air Weapons(AAW) including search radar, electronic signal monitors (ESM), fire control systems (FCS), and guns.

The measures of effectiveness (MOE) for this simulation model are: (1) the number of targets destroyed by the ship and range at which they are destroyed, (2) the number of targets striking at the ship.

These MOE's are chosen because typically, we wish to minimize the number of targets striking at the ship and maximize the ranges at which targets are destroyed. Through iterations of simulation, we can generate detailed events and states for MOE evaluation.

C. *Inductive Learning*

Inductive Learning is a method of automatically developing rules (in the form of decision trees) from example sets. Each example has a number of attributes and can be classified into a particular class. The inductive learning forms a decision tree of rules which will correctly classify all the examples on the basis of their attribute values. Quinlan's ID3 (1979), Iterative Dichotomizer 3, is the most frequently cited algorithm in inductive learning. However, it can only deal with deterministic data which could be classified purely from the attributes. This thesis will enhance ID3 algorithms to handle stochastic data which will contain a degree of random, chance variations within it as well as basic relationships between the variables. The procedures include: (1) refining input data, (2) using χ^2 significant test, instead of Quinlan's information measure, for splitting nodes on a decision tree, and (3) using post-pruning techniques to reduce the size of a decision tree.

The stochastic data from the simulation will be analyzed through the enhanced version of ID3 to obtain better weapon assignment rules. These induction rules will then be incorporated in the simulation model. The rule generation process will be terminated if the simulation results are satisfactory, otherwise the simulation and inductive learning cycle will be repeated.

D. *Objectives*

The thesis is aimed at integrating simulation modeling with inductive learning to aid the development of battle management strategies and to improve the AAW weapon effectiveness for ship air defence. We would also like to demonstrate the feasibility of combining inductive learning with simulation in producing optimized rules for air-defense.

The thesis is organized as follows: Chapter II describes the simulation model; Chapter III introduces basic algorithms in inductive learning and our modifications; Chapter IV presents the process of running simulations with rule generation and the results; in Chapter V, we summarize our research, discuss the significance of our approach, and suggest directions for future work.

II. SIMULATION MODEL

A. *Introduction*

The simulation model is a proven discrete event simulation methodology that is one of the most widely used operational research tools presently available. The methodology facilitates a top-down approach to modelling and only key features of a system are incorporated in a model.

The model processes that describe the interactions between ship and enemy targets generated by target events and their counter-actions taken by the ship are the: (1) target operation process; (2) ship motion control process; (3) detection, threat evaluation & weapon assign process; (4) fire control system process; (5) gun system process. These processes will be explained in Section E.

B. *The Definition Of Mission Attributes*

A mission attribute is a model input parameter which characterizes the operation of people and equipment as they perform a particular mission function. To obtain the value of each mission attribute, the first step is to develop criteria for evaluating each mission function. For a process level mission attribute, the criteria specify start of mission function, end of mission function, and objectives of each function. Next, the system of equipment or organization is observed in operation for a period of time. For a task each time the start of a mission occurs, the following data are collected: (1) whether or not the task objectives are achieved, (2) the length of time that is required, and (3) the environmental conditions which exist during the mission. These data are recorded in a computer data file. Each mission attribute is obtained for a collection of mission functions which are selected from the data file based on the environmental factors. Thus, a mission attribute characterizes the operation of system or organization observed under stated environmental conditions. Some mission attributes (Appendix A) are: availability, operational reliability, capability, personnel ability, and reaction time.

1. Availability

Availability is defined as a measure of the degree to which an item is in an operable and committable state at the start of a mission when the mission takes an unknown (random) time.

The definition for operational availability, A_0, is

$$A_0 = \frac{\text{Total Uptime}}{\text{Total Uptime + Total Downtime}}$$

2. Operational Reliability

Operational reliability is defined as the ability of an item to perform its required functions for the duration of a specified operational mission.

Operational reliability is expressed in the classic sense as the probability of performing an operational mission, without failure, under specified conditions. This parameter is used for systems that perform repetitive missions, such as fighters or ships. For systems or items that are used only once, e.g., "single-shot devices" such as gun systems, reliability can be expressed as a ratio of the expected number of successes to the number of total attempts. These two parameters are expressed as:

Operational Reliability = Probability of completing an operational mission of X hours without a critical failure, under specified scenario

$$\text{Probability of Success} = \frac{\text{Expected number of successful attempts}}{\text{Total number of attempts}}$$

3. Capability

Capability is summarized by the probability that the collection of equipment required to perform the process will not cause the process to fail to achieve its objectives, given that the equipment is available and operationally reliable. Some examples of capability attributes for a ship are: (1) warhead probability of kill, (2) probability that guidance subsystem will deliver the warhead within target lethality range, and (3) probability that the sensor subsystem will detect and identify the target.

4. Personnel Ability

Personnel ability is summarized by the probability that the people required to perform the process will not cause the process to fail to achieve its mission objectives, given that the equipment required is available and operationally reliable. Capability and personnel ability are assumed to be statistically independent.

5. Reaction Time

Reaction time is determined from the time of occurrence for the start of mission function, end of mission function, and objective-achieved events. For a successful process, reaction is measured from the start of test until the objective-achieved event occurs. For a failed process due to failure in operational reliability, capability, or personnel ability, reaction time is measured from the start of test until the end of test (abort) event occurs. For a process where a scarce resource has been allocated, it is important to include these abort reaction times in the model. These abort times contribute to the overall reaction time and may degrade system effectiveness.

The Gamma distribution, with the shape parameter *beta* (β) and the scale parameter *theta*(θ), has been found useful in describing typical reaction time data observed in system operation. When β is an integer, the gamma distribution having mean $\frac{\beta}{\theta}$ is related to the exponential distribution in the following manner: If the reaction time, the random variable X, is the sum of β independent, exponentially distributed random variables each having mean $\frac{1}{\theta}$, then X has a gamma distribution with parameters β and θ.

C. *Set-Up Data for the Simulation Scenario*

A particular ship and threat scenario is represented using values in the mission attribute tables (Appendix B), specified by a set of parameters including: (1) number of random variable distributions for each process, (2) number of surface ship probabilities, (3) weapon descriptions, (4) ship initial position and velocity, (5) target descriptions, (6) the data of target event program, and (7) the data of ship event program.

The random variable distributions to be used in the various processes are provided in tables

2-1, 2-2 and 2-3. Each table entry contains three values: (1) probability of an event ever occurring, (2) the shape parameter β of a Gamma distribution, and (3) mean value of the Gama distribution. In the simulation, we assume that an event will either occur (with probability 1), or not occur (with probability 0). Table 2-1 shows the detection, threat evaluation and weapon assignment random variables.

Table 2-1 Detection, Threat Evaluation & Weapon Assignment Random Variables

events & their variable	probability of an event ever occurring	the shape parameter β of a Gamma distribution	mean value $\frac{\beta}{\theta}$
Search radar Detection range (NM)	1	1	70
Reaction time from ESM detection until search radar detection (sec)	1	1	60
Reaction time from detection by search radar until begin TEWA (sec)	1	1	30
Reaction time from detection by search radar until begin valid track (sec)	1	1	6
Reaction time from start of valid track until lose valid track (sec)	0	1	120
Reaction time from lose valid track until regain valid track (sec)	1	1	10
Reaction time from regain valid track until subsequent lose valid track (sec)	0	1	150

Table 2-2 shows the fire control system random variables

Table 2-2 Fire Control System Random Variables

events & their variable	probability of an event ever occurring	the shape parameter β of a Gamma distribution	mean value (sec) $\frac{\beta}{\theta}$
Reaction time from FCS designation until the ready to fire	1	1	15
Reaction time from FCS designation until FCS fails to lockon	1	1	25
Reaction time from detection by search radar until begin valid track	1	1	25
Reaction time from FCS ready to fire until lose ready to fire and enter coast mode	0	1	200
Reaction time from FCS enter coast mode until FCS exit coast mode to either regain track or drop track	1	1	20

Table 2-3 shows the 5in, 76mm and 40mm guns system random variables.

Table 2-3 5in,76mm and 40mm Guns System Random Variables

events & their variable	probability of an event ever occurring	the shape parameter β of a Gamma distribution	mean value (sec) $\frac{\beta}{\theta}$
Reaction Time from gun assign to target until gun ready to fire	1	1	5
Reaction time from gun assign to target until abort	0	1	40
Reaction time to load gun round and fire	1	1	4
Reaction time to evaluate kill status of target at end of firing sequence	1	1	5
Reaction time to repair gun after jam occurs	0	1	240
Reaction time from gun ready to fire until lose gun synchronization with FCS	0	1	120
Reaction time from lose gun ready until regain gun synchronization FCS	0	1	180

The ship probabilities used by the processes to make decisions are provided in Table 2-4.

Table 2-4 Ship Probabilities

Probability of gun jam per round fired: 0

Probability of target kill per round fire: 0.30 (5in guns)
0.35 (76mm gun)
0.35 (40mm guns)

Probability of FCS lock on the target 1

Probability of FCS regain track or drop track,
when FCS exits coast mode 0

Probability of gun synchronization with FCS will occur or must abort gun assign,
when gun assigned to target : 1

Probability of gun to be released or wait for gun ready again,
when gun loses synchronization with FCS : 1

Table 2-5 describes the features of the weapons.

Table 2-5 Weapon Descriptions

	Availability	open fire angle	cease fire angle	magazine count
FCS-1	0.98	0	360	-
FCS-2	0.98	0	360	-
5in Gun - 1	0.90	220	140	700
5in Gun - 2	0.90	50	290	700
76mm Gun	0.95	255	105	700
40mm Gun-1	0.95	20	160	2000
40mm Gun-2	0.95	200	340	2000

The initial position X, Y, Z and velocity Vx, Vy, Vz is specified in Table 2-6:

Table 2-6 Ship Initial Position and Velocity

	X axis	Y axis	Z axis (altitude)
Position (NM)	0	0	0
Velocity (NM/hr)	20	0	0

The maximum number of targets in the scenario is specified by the parameter ATTACK-PLANE and ATTACK-MISSLE; however, the number of actual targets in the scenario can be specified by changing NTGT in BLOCK DATA or by using the interactive input routine.

The target is always specified relative to the Y axis. The target description (Table 2-7) also provides the angle of rotation of the target event program in degrees plus a X and Y translation of the target path in nautical miles. These values allow each target to approach the ship along a different threat radial with a random aim point. The random number generator used in program is RAND function in UNIX, and the SEED on calling RAND is from the number of seconds since this hour.

Table 2-7 Target Description

	Initial Position (NM)			Rotation (°)	Translation(NM)	
	X axis	Y axis	Z axis (altitude)		X axis	Y axis
Fighter	0	100	0	Random	0	0
Missile	0	27	1.3	same as above	0.1	0

The target event program (Table 2-8) contains five values for each target event: (1) target event number, (2) speed of target in nautical miles per hour, and (3) X, Y, Z position of target in nautical miles where event is to occur.

Table 2-8 Target Event Program

Fighter	Event-number		1 (Change Direction)	7 (Launch Missile)	1 (Change Direction)	1 (Change Direction)	10(End Presentation)
	Speed (NM/hr)		350	400	400	400	400
	Position	X axis	0	0	0.5	1	1
		Y axis	32	27	26	27	100
		Z axis	1.3	1.3	1.3	1.3	1.3

Missile	Event-number		4 (Seeker on)	1 (Change Direction)	6 (Begin Homing)	10 (End Presentation)
	Speed (NM/)hr		650	650	600	600
	Position	X axis	0	0	0	0
		Y axis	26	24	2.7	0
		Z axis	0.9	0.1	0.1	0

The ship event program in Table 2-9 which contains two values for each ship event: (1) ship event number, (2) slant range from ship in nautical miles where the event is to occur.

Table 2-9 Ship Event Program

Fighter	Event-number	3 (Enter ESM Area)	1 (Enter SR Area)	5 (Enter FCS Area)	2 (Exit SR Area)	12(End Presentation)
	Range (NM)	71	70	20	-70	-90

Missile	Event-number	3 (Enter ESM Area)	1 (Enter SR Area)	5 (Enter FCS Area)	9 (Enter 5in Guns Area)
	Range (NM)	71	70	20	5
	Event-number	11 (Enter 76mm Gun Area)	13 (Enter 40mm Guns Area)	15 (Target Striking at Ship)	
	Range (NM)	4	2	0	

D. *Overview of Simulation Program*

The simulation model is written in FORTRAN and is composed of a main program and 8 subprograms. Figure 2-1 shows the structure diagram of this model:

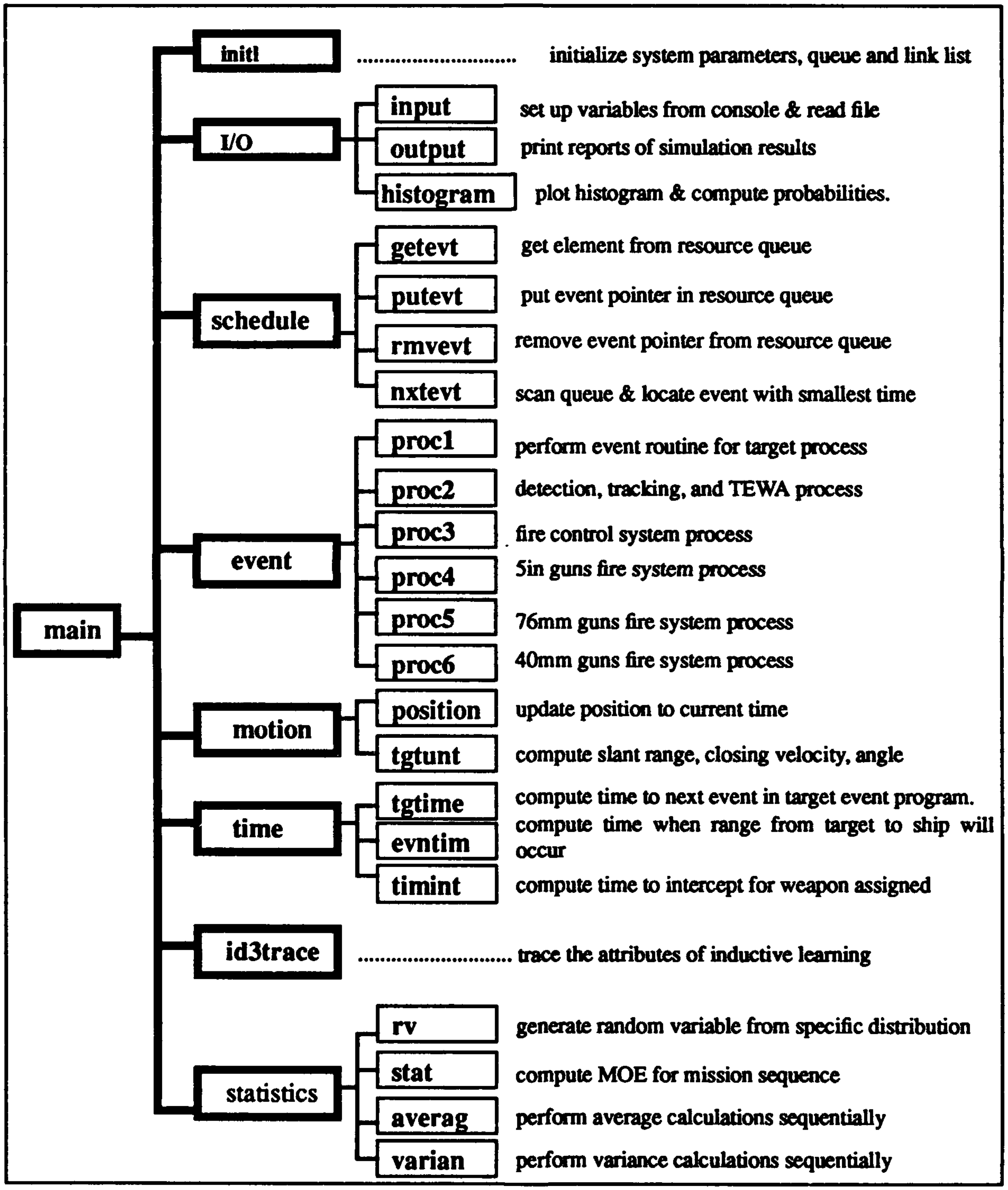

Figure 2-1 The Structure Diagram of the Simulation Model

E. Model Processes

1. Target Operation Process.

For each threat, the target operation process generates a sequence of states and events which defines the threat environment for all the other ship processes. This process schedules threat events relating to the threat offensive posture or defensive environment. These events are: (1) target change direction, (2) target seeker on, (3) target seeker off, (4) target begin homing, (5) target launches next target in sequence, and (6) end presentation.

In contrast to these events, there are offensive events that relate the threat geometry to offensive capability of the ship. These are built into the model due to their deterministic features. Moving object routines handle these events in the current simulation model.

The Beginning State (BEG) represents all time before the beginning of the presentation. The Target state (TGT) represents time between target events. When the end presentation event occurs or a target kill event occurs, process enters END state which represents the time between the end of presentation from a given threat and the end of the mission for the ship when all target presentations have ended.

2. Ship Motion Control Process.

The ship motion control process models the maneuver activities of the ship. These activities are represented in the form of maneuver command, which consists of two pieces of information: degrees to turn, and/or speed to change. The Waiting for Maneuver state (WTM) represents the time waiting for maneuver commands or the time between maneuver activities.

There are three types of maneuver command: to turn a certain degree, to change to a certain speed, and to do both at the same time. The first two types of maneuver command can be overlapped with each other, which means the commander can first order a turn, then order a speed-up later. Due to the sophisticated nature of a ship maneuver in a combined AAW scenario, a method to identify different objectives for maneuver has been devised so that each time when a maneuver is completed the program can process to achieve that specific objective.

3. Detection, Threat Evaluation and Weapon Assignment Process.

The Detection, Threat Evaluation, and Weapon Assignment (Det & TEWA) process is replicated for each threat in the scenario.

This process consists of two serially linked sub-processes: the surveillance sub-process, and tracking and TEWA sub-process. The surveillance sub-process represents the operation of the ship sensors and personnel operating them as they survey the surrounding air space searching for threats. Two types of detection sources are modelled as parallel sub-processes in the surveillance process: (1) local detection where the ship detects threats with its active radars and (2) ESM detection where the ship detects threats with passive sensors and alerts active sensor operators to their presence.

The Waiting for Surveillance states (WS1 and WS2) represent the time that a detection process waits while a threat is undetectable by active radar and passive sensors respectively. The threat is either beyond the maximum theoretical detection range of the equipment or, for active sensors, the threat is below the radar horizon. For ESM detections, the target must also be radiating as well as be within the detectable region to be detectable. Once a target detectable event occurs, the Surveillance state (SUR) for active detections or the ESM state for passive detections is entered depending on whether the target is actively or passively detectable. Each of these states represents reaction time from the time the target enters the region of detectability to the time of actual detection. Detection range for each type of detection is decided using a random variable distribution. When ESM detection occurs, the Assign Search Radar state (ASR) is entered which represents the time required for the active sensor operator to detect the target. The reaction time is decided using a random variable distribution. Once the surveillance sub-process has ended, the target is considered detected and the tracking and TEWA sub-process begins.

In the tracking sub-process, the target initially enters the Tentative Track state (TTK) when the target detection event occurs. The model assumes the use of the most capable tracking device available. When a firm track is established, the Valid Track (VTK) is entered. If track quality degenerates such that an FCS designation cannot be performed successfully then Non-Valid Track state (NVT) is entered. Thus, track quality alternates from VTK to NVT as the target approaches the ship. The duration of each state is decided using a separate random variable distribution, and a probability is used to decide if valid track is ever lost or, if lost, if valid track is ever regained. By proper selection of random variable distributions and probabilities for each target type, a sequence of track quality states, which represents actual operation, can be simulated for each threat in the scenario.

Once the target has been detected, the Evaluate state (EVL) represents the minimum time to make an evaluation to assign resources. When initial threat evaluation is complete as determined by a random variable distribution for the reaction time, threat priority is computed and the resource allocation Wait to Assign Resources state (WAR) is entered. Threat priority is set according to the closing velocity and range of the target to ensure the highest priority is assigned to targets requiring minimum intercept time. Threat priority is continuously updated as the target approaches the ship. The target remains in the wait for assign resource state (WAR), and resources such as FCS, and GUN are assigned when the target comes within assignable range and the target has the highest priority.

4. Fire Control System Process

The fire control system process represents the operation of a fire control channel and its operator, starting from FCS designation through the lock on event and tracking of the target. This process is duplicated for each FCS on the ship.

The initial Wait for Designation state (WDS) represents idle time while the FCS is waiting for target designation. When a FCS is designated, a probability is used to decide if lockon occurs or not. If no lockon occurs, the Abort state (ABT) is entered which represents the time spent attempting to achieve a lockon before the designation is aborted. Then the abort FCS designation event occurs, the state changes to the WDS state and the FCS is released to be assigned to another target or the same target.

If a lockon occurs, then the FCS state is entered which represents the time required to achieve the lockon condition. When the event of FCS ready to fire occurs, the FCT state is entered which represents the time that the FCS is tracking the target. a random variable distribution is used to decide how long the FCS will track the target before going into coast mode. Coast mode is represented by the Fire Control Coast state (FCC). It is a period of operation where the FCS continues to move the tracking antenna using previous tracking rates in the hope that the video will reappears. The period allowed for coast mode is decided using a random variable distribution. When the event of FCS exit coast mode occurs, a probability is used to decide if FCS track is regained or the track is dropped. If the track is regained, the Fire Control Track state (FCT) is entered; else, the drop state occurs. The Drop FCS track state (DRP) occurs for a fixed time then the FCS is released and the WDS state is entered.

5. Gun System Process.

The gun system process represents operations of a gun system and its operators. The gun is synchronized with a FCS to fire a round. The gun system process is duplicated for each gun system on the ship.

The gun system is mainly assigned to surface targets and closing air targets. Gun assignment occurs whenever the FCS is locked on the target. When the gun is assigned to a target, a probability is used to decide if gun synchronization will occur or not. If not, then the ABT state is entered to represent the time that synchronization is attempted but not achieved. When the event of abort gun system assignment occurs, the state is changed to Wait for Gun Assign state (WAG) and the gun is released to be assigned to another target or the same target depending on relative target priorities.

If gun synchronization occurs, then the Gun System Synchronization state (GSS)is entered which represents the time required to achieve the state of gun system ready to fire. When a gun is ready to fire, the gun system process splits into two parallel sub-processes: the top process models the firing sequence and the bottom process models gun synchronization.

The initial state of the top process is Fire Logic State (FLG) which represents the time that the gun system waits for the open fire logic to be stultified. If the target is within maximum gun open fire range for the ship, the gun ready to fire condition exists, and the FCS is still tracking when the FLG state is entered then a firing can occur. At gun open fire, the Fire state (FIR) is entered which represents the firing time. It is assumed that once the gun is assigned and the firing sequence begins then the gun will continue to fire rounds until the target is destroyed or the gun jams or runs out of ammunition, or the target goes out of gun range. These outcomes are represented by the Kill (KIL), Jam (JAM), and Evaluation (EVL) states; respectively.

The initial state of the bottom process is the Ready state (RDY) which is the ready to fire condition. The gun system will remain in the RDY state until synchronization is lost which is decided using a random variable distribution for the RDY time. When the event of lose ready to fire occurs, a probability is used to determine if the gun system will be released or if the ship will wait until gun system synchronization is regained. If the ship decides to wait then the Gun Not Relay state (GNR) is entered, else the RGN state is entered. A random variable distribution is used to compute the duration of the gun not ready condition. The release time is fixed in the program. When the gun is released the WAG state is entered.

III. INDUCTIVE LEARNING

A. *The Bayesian Rule*

Inductive learning is a method for automatically generating a decision tree from a learning sample which consists of data $(x_1, j_1), \ldots, (x_N, j_N)$ on N cases, where $x_i \in \mathcal{X}$ ($\mathcal{X}$ is the set of all possible attribute vectors which can be classified into a particular class, where each attribute can be integer valued or symbolic), $j_i \in C = \{1, \ldots, J\}$ where C is the set of classes, and $i = 1, \ldots, N$. In other words, the learning sample is denoted by the set $S = \{(x_1 j_1), \ldots, (x_N, j_N)\}$.

A systematic way of predicting class membership is a rule that assigns a class membership in C to every measurement vector x in $\mathcal{X}$.

The major guide that has been used in inductive learning is the Bayesian rule. To make the concept precise, a probability model is needed. Define the sample space as a set of all couples (x, j) where $x \in \mathcal{X}$ and $j \in C$. Let P(A, j) be a probability on the sample space $\mathcal{X} \times C$, $A \subset \mathcal{X}$, $j \in C$. The interpretation of P(A, j) is that a case drawn at random from the relevant population has probability P(A, j) that its attribute vector x is in A and its class is j. Assume that the learning sample S consists of N cases $(x_1, j_1), \ldots, (x_N, j_N)$ independently drawn at random from the distribution P(A, j). Construct an induction rule $d: \mathcal{X} \to C$ using the sample S.

Let (x, y), $x \in \mathcal{X}$, $y \in C$, be random, taken from the probability distribution *P(A, j);* i.e., $P(x \in A, y = j) = P(A, j)$, where (x, y) is independent of S.

Then $d_B(x)$ is a Bayesian rule with a zero-one loss function if for any other induction rule d(x),

$$P(d_B(x) \neq y) \leq P(d(x) \neq y)$$

The Bayesian misclassification rate is

$$R_B = P(d_B(x) \neq y).$$

To illustrate how $d_B(x)$ can be derived from P(A, j), we give its form in an important special case, and we define the prior class probabilities $\pi(j)$, j = 1,..., J, as

$$\pi(j) = P(y = j)$$

and the probability distribution of the jth class attribute vectors by

$P(A|j) = P(A, j) / \pi(j)$.

Assume $\mathcal{X}$ is the M-dimensional euclidean space and for every j, j = 1,..., J, P(A|j) has the probability density $f_j(x)$; i.e., for sets $A \subset \mathcal{X}$,

$$P(A|j) = \int_A f_j(\mathbf{x})\, d\mathbf{x}$$

Then, the Bayesian rule with respect to the loss function $L(d(x),y) = 1, \quad d(x) \neq y,$

$L(d(x),y) = 0, \quad d(x) = y,$

is defined by

$$d_B(x) = j \text{ on } A_j = \{x;\ f_j(x)\,\pi(j) = \max\,[f_i(x)\pi(i)]\},\ \forall i \tag{3.1}$$

and the Bayesian misclassification rate is

$$R_B = 1 - \int max\,[f_j(x)\pi(j)]\, dx,\ \forall j$$

Although d_B is called the Bayes rule, it is also reorganized as a maximum likelihood rule: Classify $\mathbf{x}$ as that j for which $f_j(x)\pi(j)$ is maximum. As a minor point, note that (3.1) does not uniquely define $d_B(x)$ on points $\mathbf{x}$ such that $\max[f_j(\mathbf{x})\,\pi(j)]$ is achieved by two or more different j's. In this situation, define $d_B(x)$ arbitrarily to be any one of the maximizing j's.

B. *Construction of The Decision Tree*

The decision tree is constructed through repeated splits of subsets of $\mathcal{X}$, beginning with $\mathcal{X}$ based on a learning sample S into descendant subsets. When $\mathbf{x}$ finally moves into a terminal subset, its predicted class is given by the class label attached to that terminal subset.

In other words, the entire construction of a tree, revolves around three elements:

1. The selection of splits.
2. The decision about when to declare a node terminal or to continue splitting it.
3. The assignment of each terminal node to a class.

The crux of the problem is how to use the sample data S to determine the splits, the terminal nodes, and their assignments. It turns out that the class assignment problem is simple. The critical point is to find good splits and to know when to stop slitting.

The first problem in tree construction is how to use sample S to determine the splits of $\boldsymbol{X}$ into smaller and smaller pieces. The fundamental idea is to select each split of a subset so that the data in each of the descendant subsets are "purer" than the data in the parent subset. The idea of finding splits of nodes so as to give "purer" descendant nodes was implemented in this way:

1. For a node t, define the node proportions p(j|t), j = 1,..., n, to be the proportion of the cases $x_n \in t$ belonging to class j, so that p(1|t) +... + p(n|t) = 1.
2. Define a measure i(t) of the impurity of t as a nonnegative function ϕ of the p(1|t),..., p(n|t) such that

$$\phi\ (1/n,\ 1/n,\ldots,\ 1/n) = \text{maximum},$$

$$\phi\ (1, 0,\ldots, 0) = 0,\ \ \phi\ (0, 1, 0,\ldots, 0) = 0,\ldots,\ \ \phi\ (0,\ldots, 0, 1) = 0$$

ϕ is a symmetric function of p(1|t)..., p(n|t).

That is, the node impurity is largest when all classes are equally likely, and smallest when the node contains only one class.

For any node t, suppose that there is a candidate split k of the node which divides it into t_L and t_R such that a proportion P_L of the cases in t go in t_L and a proportion P_R go into t_R (Figure 3-1.

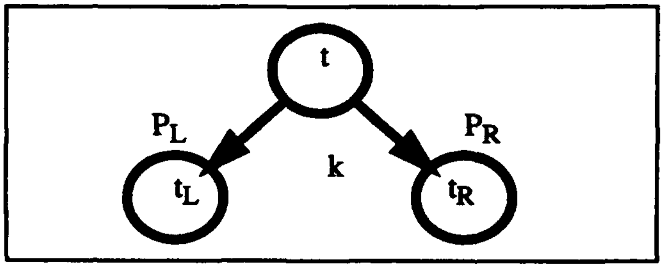

Figure 3-1 Node t Split Into node t_L and t_R

Then the goodness of the split is defined to be the decrease in impurity

$$\Delta i(k, t) = i(t) - P_L i(t_L) - P_R i(t_R)$$

The final step is:

3. Define a candidate set K of splits k at each node. Generally, it is simpler to conceive of the set S of splits as being generated by a set of questions Q, where each question in Q

is of the form is "$x \in A?, A \subset \mathcal{X}$".

Then the associated split k sends all $\mathbf{x}_n$ in t that answer "yes" to t_L and all $\mathbf{x}_n$ in t that answer "no" to t_R.

Now, the node impurity is defined as

$$i(t) = -\sum_{1}^{n} p(j|t) \log p(j|t)$$

There is no convincing justification for this specific form of i(t). It was selected simply because it was a familiar function having the properties required by step 2.

The tree is grown in the following way: at the root node t_1, a search is made through all candidate splits to find that split k* which gives the largest decrease in impurity; i.e.,

$$\Delta i(k^*, t_1) = \max \Delta i(k, t_1), k \in K$$

Then t_1 was split into t_2 and t_3 using the split k* and the same search procedure for the best $k \in K$ repeated on both t_2 and t_3 separately.

To terminate the tree growing, the following heuristic rule is used: When a node t cannot be split without significantly decreasing impurity, then t becomes a terminal node.

The class character of a terminal node was determined by the plurality rule. Specifically, if

$$p(j_0|t) = \max(p(j|t)), \forall j$$

then t was designated as a class j_0 terminal node.

In summary, the tree-constructing procedure proceeds as follows:

(1) Use some heuristic (*the splitting heuristic*) to find the attribute vector $\mathbf{x}$ whose values best split the learning sample S into their class set j. Numerical attributes are split at some threshold value of the attribute with each possible threshold being tested to find the best.

(2) Partition the sample S according to the value of the chosen attribute vector $\mathbf{x}$. Splits on numerical attributes are always binary. Consequently, they can be re-used as there may still be sample in the cases partition with different values for that attribute.

(3) Recursively apply (1) and (2) on each partition until the termination heuristic applies.

When all branches are established, then the induction procedure is complete, and we have a decision tree which can be used to classify new instances.

C. *Using The ID3 Algorithm*

Quinlan's ID3 (1979), the Iterative Dichotomizer 3 algorithm is the most frequently cited algorithm for inductive learning. Implementations of ID3 vary, but the principle is that the algorithm induces a decision tree of rules from a learning sample, which consists of classes and attributes. This learning sample can be set up in a contingency table as shown in Table 3-1:

Table 3-1 Contingency Table of Learning Sample

		Class					
		C_1	C_2	. . .		C_j	Total
	A_1	x_{11}	x_{12}	. . .			$x_{1\cdot}$
Attribute	A_2	x_{21}	x_{22}	. . .			$x_{2\cdot}$
	.	.	.				:
values	.	.	.				.
	.	.	.		x_{mn}		.
	A_i					x_{ij}	$x_{i\cdot}$
Totals		$x_{\cdot 1}$	$x_{\cdot 2}$	. . .		$x_{\cdot j}$	N

where x_{mn} is the number of examples in class n with attribute value m, N is the sample size. The information content of the class total as a whole is

$$M(C) = -\frac{x_{\cdot 1}}{N}\log\frac{x_{\cdot 1}}{N} - \frac{x_{\cdot 2}}{N}\log\frac{x_{\cdot 2}}{N} - \ldots$$
$$= -\frac{1}{N}\sum x_{\cdot n}\log x_{\cdot n} + \log N$$

The information content for the row A_1 is

$$M(A_1) = -\frac{x_{11}}{x_{1.}}\log\frac{x_{11}}{x_{1.}} - \frac{x_{12}}{x_{1.}}\log\frac{x_{12}}{x_{1.}} - \ldots,$$

and similarly for the other values of A.

Taking an average of these A values weighted by the frequency of occurrence of each (the row totals) gives

$$B(C|A) = \frac{x_{.1}}{N}M(A_1) + \frac{x_{.2}}{N}M(A_2) + \ldots$$

$$= -\frac{1}{N}\left(\sum\sum x_{mn}\log x_{mn} - \sum x_{m.}\log x_{m.}\right)$$

The information measure is then defined as the *"gain"* in information brought about by knowledge of the attribute

$$IM = M(C) - B(C|A)$$

$$= \frac{1}{N}\left(\sum\sum x_{mn}\log x_{mn} - \sum x_{m.}\log x_{m.} - \sum x_{.n}\log x_{.n}\right) + \log N$$

Then the ID3 algorithm works as follows:

(1) Take each attribute in turn, and calculate an information measure of how well the values of the attribute split the data into their classes.

(2) Choose the attribute that has the maximum information measure and partition the learning sample according to the test based on the value of this attribute. Let R be a test specified completely by the mutually exclusive and exhaustive set of possible outcomes $\{R_1, R_2, \ldots, R_i\}$ that it might have. For a continuous attribute A we consider only tests of the form $\{"A \le T", "A > T"\}$ for some real threshold T; for an attribute A with discrete possible values $V_1, V_2, \ldots, V_i$ we consider tests of the form {"A=V_m", m =1, 2,..., i,}. Whatever the form of the test, each case in S can have only one of the possible outcomes, so R partitions S into subsets $\{S_1, S_2, \ldots, S_i\}$, where S_m consists of those cases that have outcome R_m.

(3) For each partitioned subset, repeat steps 1 and 2 until each case in the learning sample

is correctly classified or no more attributes are available because they have all been used along a particular branch.

(4) Finally, the resulting structure is a decision tree for the original learning sample S.

D. *Using The χ^2 Test And Post-pruning Decision Tree*

1. The χ^2 Test

The χ^2 test, with test statistic :

$$x^2 = \sum\sum\frac{(x_{mn} - E_{mn})^2}{E_{mn}}$$

where $E_{mn} = \frac{x_{m.}x_{.n}}{N}$ tests the null hypothesis that the rows and columns of a contingency table are independent.

The χ^2 test has previously been suggested by Quinlan and others for use in noisy domains. Noise means contradictions, i.e. conflicting examples which have identical (or similar in the case of numerical data) sets of attribute values but are differently classified. In this case, a probabilistic approach to classification is adopted. When a subset of a learning sample contains only contradictory cases, the most popular class present in the subset may be assumed to be the correct class; alternatively, a probability is assigned to each class, according to how often it appears at that node. This approach could lead to unjustifiably large trees, since there may be no information to be gained by further testing of attributes before a node containing only contradictions is found. This would be the case when all attributes remaining were irrelevant. The χ^2 test can detect these cases when there is no significant correlation and the induction process can be halted. Stochastic data suggests probabilistic classification, since it will not build branches which are not statistically valid; this has the added advantage that it usually leads to a much smaller tree.

Thus the χ^2 test appears well suited to our purpose. It generates a compact probabilistic tree, and identifies the most statistically significant attributes.

However, it imposes additional limitations upon our data in order to maintain its statistical validity. In order to generate a branch, all of the following must be satisfied:

(1) The total frequency must be at least 50.

(2) Attributes with different numbers of values cannot be mixed since they have different degrees of freedom. A violation would render the comparison of the χ^2 values invalid. This is not a problem in this thesis since we have only integer attributes, and as the attributes are always split into two subsets at a threshold value, effectively they always have one of the two values, depending on which side of the threshold they fall.

(3) When the contingency table for an attribute is drawn up showing the number of instances of each class for each value of the attributes, the expected value should be at least five for each value m and class n (i.e. $E_{mn} \geq 5$).

(4) Finally, in order to generate a branch, the χ^2 value for the appropriate number of degrees of freedom must be significant at the chosen confidence level. Lowering the required significance level will lead to a larger tree.

2. Post-pruning the Decision Tree

For post-pruning of the decision tree, the tree is first fully developed and then pruned back from the leaf nodes to the first significant branch. Relaxing the $E_{mn} \geq 5$ constraint and generating branches for which the χ^2 statistic are not significant at the specified level will lead to a much larger tree before pruning. Post-pruning is useful for finding 'local' correlations between attributes and classes, which would otherwise be missed. For instance, there may be an attribute which discriminates well between two classes, but before it can be used, none of the other classes is present in that subset; thus the $E_{mn} \geq 5$ test is failed, and the induction is stopped despite the fact that the branch would be significant.

E. Using Pop-11 Language

There are three major steps in Inductive Learning: (1) generating ID3 decision tree, (2) generating χ^2 test and post-pruning decision tree, and (3) converting decision tree to FORTRAN code so that it can be linked to simulation program. The ID3 and χ^2 decision tree are generated by

"Rule Induction Software" written by P. R. Race and R. C. Thomas (1988).

All programs except the simulation are written in POP-11 which is an interactive language, has a structure like PASCAL, storage allocation like LISP, and argument passing mechanism like FORTH. POP-11 syntax is based on blocks, each of which has matching opening and closing keywords. The basic functional unit is called *procedure*. There are a great number of built-in procedures for common used operations. The user can also define their own procedures as necessary. Built-in and user-defined procedures can mix freely in a program.

There are many pre-defined data types in POP-11. The simplest is numeric type. There are also vectors and arrays for aggregated data, boolean to indicate binary truth values, and lists of the same kind as in LISP and PROLOG. In addition, procedures are also just another kind of data structure and can be manipulated and passed around as parameters as easily as any other data types.

One of the most significant aspects about POP-11 is dynamic storage allocation. This means that data structures are built at run-time and a programs need never worry about how much space to allocate for certain resources. The system looks after all the memory management, and has a process called garbage collection to clean up unused space whenever needed. This contrasts sharply against most other conventional programming languages.

POP-11 is a loosely-typed language. That is, procedures do not have to restrict their arguments and results to be of a single type. Also, the elements of a compound structure (such as an array) do not all have to be of the same type. This is extremely important. It means that when designing a program, we can match the representation in the program very closely to the object it represents. It also means we can use the same procedures for manipulating different kinds of object. Consequently, POP-11 will not be able to tell the type of an object merely by noting where it comes from, and a hidden portion of each object is used to describe what kind of object it is.

All data, on its way from A to B, passes through a holding bay called the *stack*. This stack is just like the one in FORTH, except that programs can find out what kind of object each item is as it comes out of the stack, rather than relying on the structure of the program, as in FORTH. The stack works like this: you can push objects onto it, and pop them back off later. The first object on is the last object off. Most of the time, the operation of the stack is not something which needs thinking about.

However, POP-11 was originally developed as a language for AI applications. It is the core language of the POPLOG system. Within POPLOG, Prolog, Common Lisp, and POP-11 and ML are incrementally compiled into POPLOG Virtual Machine (VM) code which is then compiled in efficient binary code for the host machine. At the VM level, the POPLOG languages share the same instruction set and data types, allowing very tight integration; code written in different languages can easily be combined in user programs. In addition, code written in any non-POPLOG language which compiles to a standard object module - such as C, Pascal, Fortran, Ada, etc. - can be linked into a POPLOG application and called directly from Prolog, Lisp, POP-11, or ML with unrestricted passing of arguments and results.

In this study, we use Fortran in simulation model and POP-11 in inductive learning, then simulation (deduction) and inductive learning can be linked together, and recursively work with loop circle.

IV. DATA PROCESSING AND OUTPUT ANALYSIS

A. *Selection of Learning Samples*

1. Structured Approach for Large Rules

There are many rules in the simulation model for system operation. One important rule is about weapon assignment during engagement. The features chosen from weapon assignment and their values traced during simulation form the attributes to be used in the inductive learning algorithm. If useful features are excluded, it may be impossible to find a consistent classifying rule, or the induction rule may be too large and become a poor classifier of new test cases. One method used in this study is to break a large and complex weapon-assignment rule into smaller and more manageable sub-rules. This is a much better approach than constructing one very large learning sample where some attribute values may be irrelevant or redundant.

Figure 4.1 shows the structure for dealing with weapon assignment for different sub-rules with their attributes. We select 5in Guns, 76mm Gun and 40mm Guns sub-rules (Appendix C) and generate random samples by tracing their attribute values during simulation.

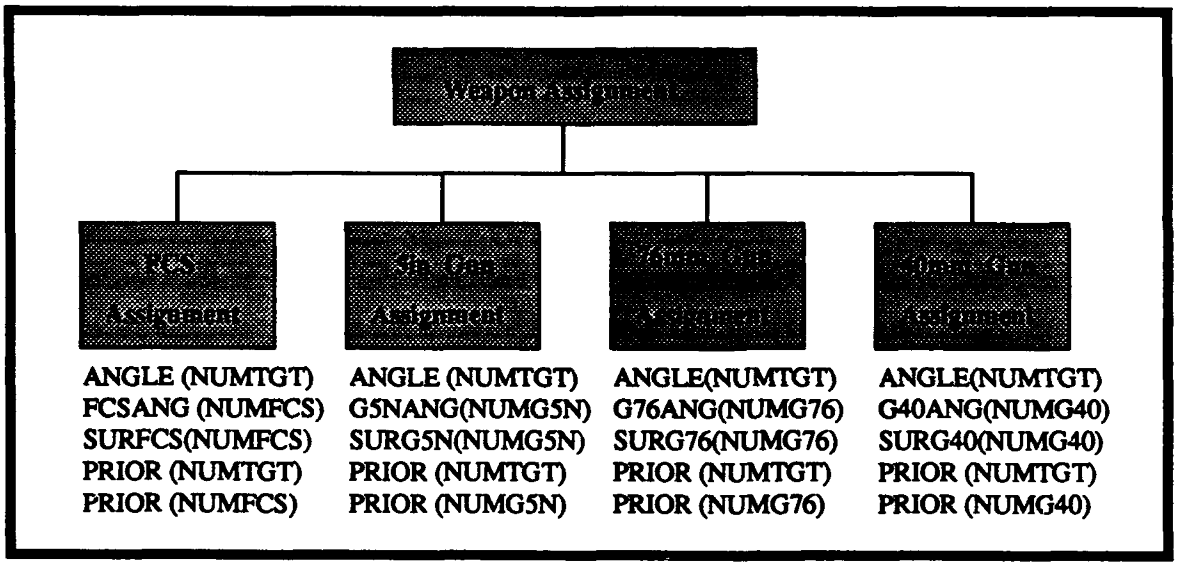

Figure 4-1 The Structure of Weapon Assignment with Their Attributes

2. Saturation Attack for Learning Samples

Simulations will start when the fighter aircraft is within 100 nautical miles from ship. Figure 4-2 shows the target detection, threat evaluation and weapon assignment area around the ship in defense position.

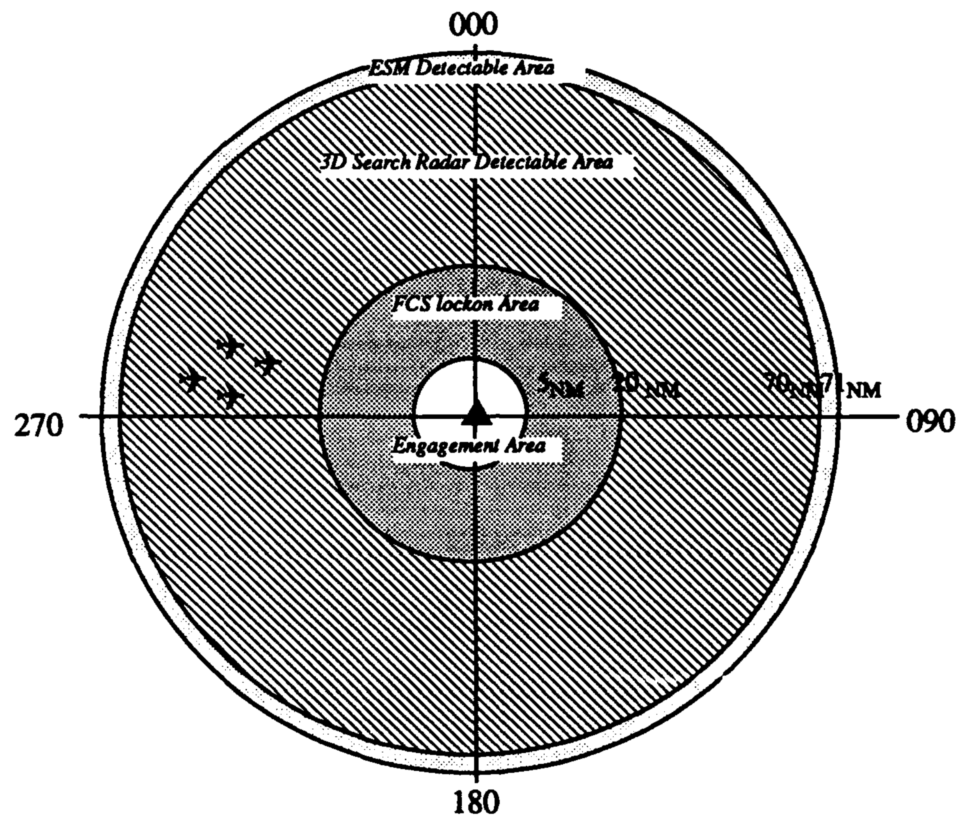

Figure 4-2 Target Detection, Weapon Assignment and Engagement Area

The fighter aircraft randomly comes from any direction. After it launches a missile 27 NM from the ship, it will leave. The missile closes on to the ship, and then is followed by homing at the distance of 2.7 NM. We assume the homing probability is 1. Figure 4-3 shows the ships engagement area with different guns.The fighter aircraft can continuously launch missiles at the interval of 0.1 NM. In these experiments, three and four missiles are used to run the simulation and gauge the target saturation attack. Table 4-1 shows that the target saturation attack for one fighter aircraft is with 4 missiles.

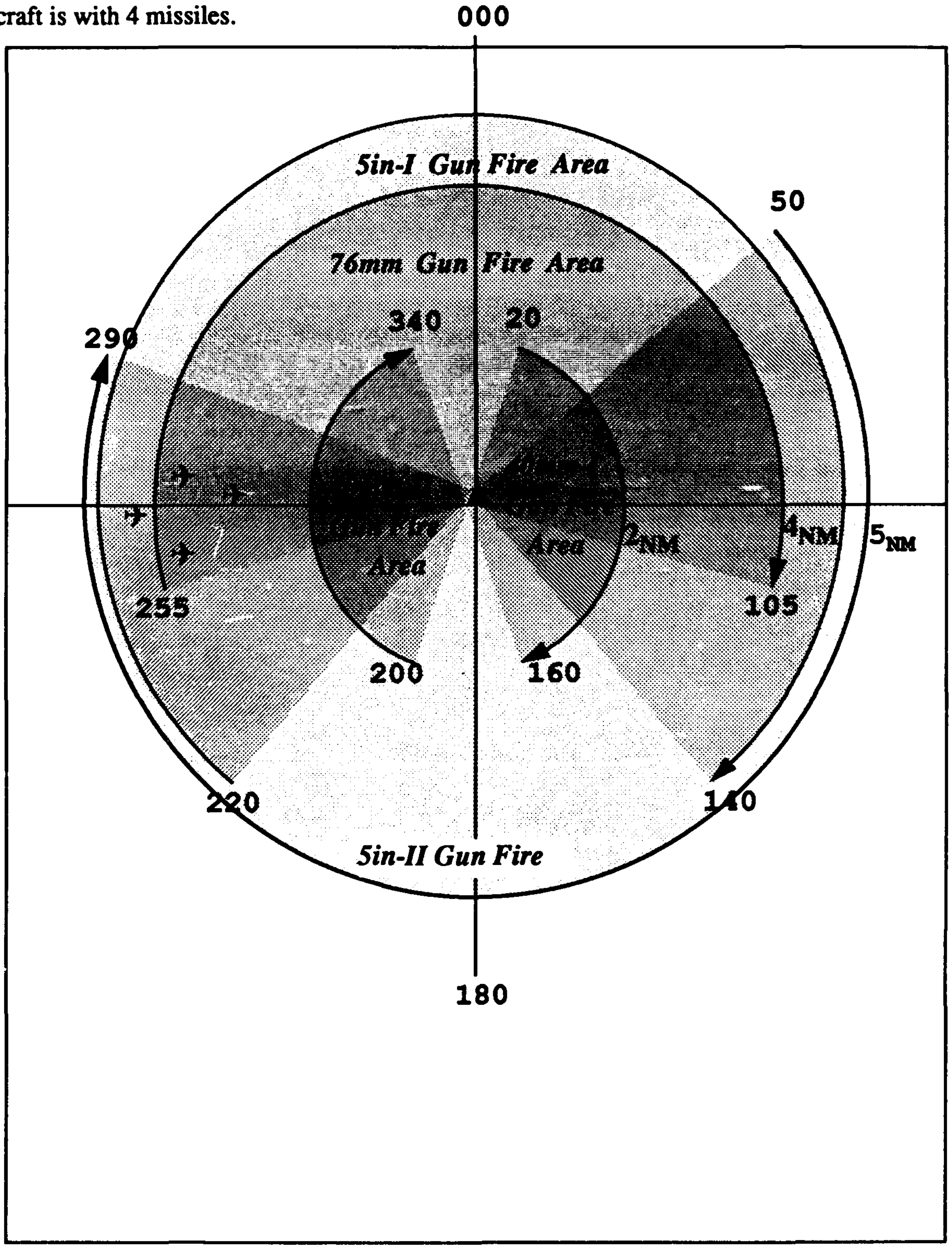

Figure 4-3 Ship Engagement Area with Different Guns

Table 4-1 Target Saturation Attack for 1 Fighter with 4 Missiles

NUMBER OF MISSILES STRIKING SHIP (**1 fighter with 3 missiles**)

MEAN=1.4660, STDEV=0.8365, N= 2458

NO MISSILE	**	193/2458_____ 8%
1 MISSILE	*************	1285/2458_____52%
2 MISSILES	******	620/2458_____25%
3 MISSILES	****	360/2458_____15%

NUMBER OF MISSILES STRIKING SHIP (**1 fighter with 4 missiles**)

MEAN= **2.4778,**STDEV= 0.8192, N= 2496

NO MISSILE		1/2496_____ 0%
1 MISSILE	**	165/2496_____ 7%
2 MISSILES	*************	1323/2496_____53%
3 MISSILES	*******	652/2496_____26%
4 MISSILES	****	355/2496_____14%

Prob (number of missiles striking at ship > 0 | three missiles) = 0.92
Prob (number of missiles striking at ship > 0 | four missiles) = 1
Therefore, 4 missiles is the target saturation attack for 1 fighter aircraft

We ran the simulation with 4 missiles on every 0.3° around the ship, so that the sample is large enough to be representative for inductive learning. Table 4-2 summarizes the learning sample generated by the simulation program, and formatted with POP-11 language with different attribute values and classes (details are as Appendix D).

Table 4-2 The Summary of the Learning Sample for Weapon Assignments

	Attributes						Classes
	SURG5N (gun #1)	SURG5N (gun #2)	Engagement Angle	PRIOR (# target)	PRIOR (gun #1)	PRIOR (gun #2)	
5in Guns (988 trials)	"S1"(ready to fire)	"S1"(ready to fire)	0°~360°	0~# target	0, 1, 2	0, 1, 2	LOGIC=0 (not assign)
	"S2"(gun synchronization)	"S2"(gun synchronization)					LOGIC=1 (assign gun #1)
							LOGIC=2 (assign gun #2)
	Attributes						Classes
	SURG76 (gun #1)	Engagement Angle	PRIOR (# target)	PRIOR (gun #1)			
76mm Gun (540 trials)	"S1"(ready to fire)	0°~360°	0~# target	0, 1			LOGIC=0 (not assign)
	"S2"(gun synchronization)						LOGIC=1 (assign gun #1)
	Attributes						Classes
	SURG40 (gun #1)	SURG40 (gun #2)	Engagement Angle	PRIOR (# target)	PRIOR (gun #1)	PRIOR (gun #2)	
40mm Guns (328 trials)	"S1"(ready to fire)	"S1"(ready to fire)	0°~360°	0~# target	0, 1, 2	0, 1, 2	LOGIC=0 (not assign)
	"S2"(gun synchronization)	"S2"(gun synchronization)					LOGIC=1 (assign gun #1)
							LOGIC=2 (assign gun #2)

B. *Refining the Learning Samples*

The quality of the learning samples influence the quality of the induction tree. However, before the results of induction are satisfactory, the learning samples have to be refined. Appendix E shows the conflicting cases picked up from the learning samples. Those cases have identical (or similar in the case of numerical data) sets of attribute values but are differently classified.

C. *Induction on the Learning Samples*

Having constructed the learning samples, the induction of decision rules were carried out by the ID3 algorithm and the χ^2 algorithm using "Rule Induction Software". The resulting decision tree of the ID3 algorithm is too large to be easily comprehensible, but a smaller and more efficient decision tree is generated by the χ^2 algorithm. Appendix F shows the induction rules generated from the ID3 and the χ^2 algorithms. For example, the induction rules for 76mm gun assignment are:

```
SURG76(1,1)
S1: ANGLE
    < 255: ANGLE
          < 104:  LOGIC = 1
          >= 104: LOGIC = 0
    >= 255: LOGIC=1
S2: LOGIC=0
```

This means when SURG76(gun number 1) = "S1" (ready to fire), and engagement angle < 104°, then LOGIC = 1 (76mm gun number 1) will be assigned. The remainder of this appendix can be explained the same way. Table 4-3 shows their comparison. Clearly, the χ^2 algorithm generates a more compact decision tree.

Table 4-3 Comparison of the Decision Tree for Different Induction Rule Algorithm

Weapons	Sample Size	# of Attributes	# of Classes	# of Nodes	Rule Algorithms
5in Guns	988	6	3	137	ID3
				21	χ^2
76mm Guns	540	4	2	9	ID3
				7	χ^2
40mm Guns	328	6	3	19	ID3
				19	χ^2

D. *Post-Pruning the Induction Tree*

If the learning samples have conflicts (identical or similar attribute values but different classes) that are left unrefined, the χ^2 algorithm will produce sub-trees that attempt to resolve these conflicts. One resolution is to allow the χ^2 algorithm to first build a tree, then to prune it by replacing sub-trees with the first significant leaf. Figure 4-4 shows the results of pruning the induction tree of the 5in Guns assignment. The shaded area represents the part of induction rules of original simulation results when the engagement angle is greater than or equal to 51°. It occurs 31 times when the 5in guns are not assigned, 169 times when 5in gun number 2 is assigned, and 0 times when 5in gun number 1 is assigned. Since in this case, gun number 2 has a much greater probability of being engaged, it is used to represent this branch of the tree. In other words, the shade part is pruned back from the leaf nodes to the most significant branch, and "LOGIC = 2" (gun number 2) is now used to represent the shaded part

```
PRIOR(TGTG5N(1))
< 1: SURG5N(1,2)
        S1: PRIOR(NUMTGT)
              < 3: ANGLE
                     < 220: ANGLE
                               < 140: LOGIC=1 197
                              >= 140: LOGIC=2 121
                    >= 220: LOGIC=1 226
             >= 3: LOGIC=0 1
        S2: LOGIC=0 75
>= 1: ANGLE
    < 289: ANGLE
              < 51: SURG5N(1,1)
                       S1: LOGIC=1 8
                       S2: LOGIC=0 50

             >= 51: [illegible]          LOGIC = 2

    >= 289: SURG5N(1,1)
                 S1: LOGIC=1 3
                 S2: LOGIC=0 73
```

Figure 4-4 Post-Pruning 5in Guns Induction Rules

E. *Transforming Induction Results into Simulation Model*

After pruning the induction tree, we have the refined induction rules for weapon assignment . These rules were then translated into Fortran code (see Appendix G) and incorporated in the simulation model. (The translator was written in POP-11, as Appendix H)

F. *Output Analysis for Simulation Model*

1. Comparing Two Rules Using Random Samples

Using two different rules (the original rule and the induction rule), we ran the simulations based on the stopping criterion of 95% confidence limit for all the MOEs. Table 4-4 shows the results of hypothesis testing for the simulation runs of saturation attack (1 fighter with 4 missiles) using two different weapon assignment rules. It was found that the induction rules produced significant improvements in reducing the expected number of missiles striking at ship and in increasing the expected number of missiles destroyed by 5in Guns. This means that the result of the simulation using induction rules have better MOE's in terms of minimizing the number of missiles striking at the ship and maximizing the ranges of target destroyed.

Table 4-4 Hypothesis Test for Significant Difference of Two Simulation Results

Simulation (1 fighter and 4 missiles) stopped until all the MOEs met the 95% confidence interval.				
	Expected # missiles striking ship		Expected # missiles destroyed by 5in Guns	
	Original Rules	Induction Rules	Original Rules	Induction Rules
Simulation Results	$\overline{X_1}$=2.4778 n_1=2496 s_1=0.8192	$\overline{X_2}$=2.3328 n_2=2770 s_2=0.6879	$\overline{X_1}$=1.1830 n_1=2496 s_1=0.7673	$\overline{X_2}$=1.3480 n_2=2770 s_2=0.6697
Hypothesis	$H_0 : \mu_1 = \mu_2$ $H_a : \mu_1 > \mu_2$		$H_0 : \mu_1 = \mu_2$ $H_a : \mu_1 < \mu_2$	
Computation	$Z = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$ = 6.8801 > 1.96 ($Z_{.025}$)		$Z = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$ = -8.2298 < -1.96 ($-Z_{.025}$)	
Conclusion	Reject H_0, accept H_a. Therefore, induction rules have significant improvements to reduce the expected number of the missiles striking at ship		Reject H_0, accept H_a. Therefore, induction rules have significant improvements to increase the expected number of the missiles destroyed by 5in Guns	
Remarks: X = Sample mean of expected value, s = sample standard deviation of expected value, n = sample size, all the hypothesis tests with critical value α=.025 (95% confidence)				

2. Comparing Two Rules Using All Possible Engagement Angle Cases

In this experiment, We ran the simulation model 20 times for each possible engagement angle (1°) using 2 attack missiles to 5 attack missiles. Table 4-5 shows the summary of the results for two different rules when saturation attack (4 attack missiles) occurs. Figure 4-5, 4-6, 4-7, and 4-8 (number of attack missile from 2 to 5) shows the comparison of the number of missiles striking at ship with two different rules. It is clear that simulation using induction rules have better MOE in terms of minimizing the number of missiles striking at ship.

Table 4-5 The Summary of the Simulation Result for Two Different Rules

Simulation model runs 20 times at each possible engagement angle (1°) with 1 attack fighter and 4 attack missiles		
Expected # of missiles destroyed by 5in Guns	Original Rules	MEAN = 1.1828, STDEV = 0.7690, N = 7200 no missile |***************** 1590/N .... (22%) 1 missile |******************************* 2703/N .... (38%) 2 missiles|******************************** 2907/N .... (40%)
	Induction Rules	MEAN = 1.3416, STDEV = 0.6796, N = 7200 no missile |********** 852/N .... (12%) 1 missile |*********************************** 3035/N .... (42%) 2 missiles|************************************* 3313/N .... (46%)
Expected # of missiles destroyed by 76mm Guns	Original Rules	MEAN = 0.1786, STDEV = 0.4510, N = 7200 no missile |******************************* 6118/N .... (85%) 1 missile |**** 878/N .... (12%) 2 missiles|* 204/N .... (3%)
	Induction Rules	MEAN = 0.1765, STDEV = 0.4504, N = 7200 no missile |******************************* 6136/N .... (85%) 1 missile |**** 857/N .... (12%) 2 missiles|* 207/N .... (3%)
Expected # of missiles destroyed by 40mm Guns	Original Rules	MEAN = 0.1661, STDEV = 0.3788, N = 7200 no missile |******************************** 6022/N .... (84%) 1 missile |***** 1160/N .... (16%) 2 missiles| 18/N .... (0%)
	Induction Rules	MEAN = 0.1585, STDEV = 0.3723, N = 7200 no missile |******************************** 6078/N .... (84%) 1 missile |***** 1103/N .... (15%) 2 missiles| 19/N .... (0%)
Expected # of missiles striking ship	Original Rules	MEAN = 2.4720, STDEV = 0.8350, N = 7200 no missile | 3/N .... (0%) 1 missile |***** 528/N .... (7%) 2 missiles|** 3807/N .... (53%) 3 missiles|******************** 1789/N .... (25%) 4 missiles|************ 1073/N .... (15%)
	Induction Rules	MEAN = 2.3229, STDEV = 0.7066, N = 7200 no missile | 2/N .... (0%) 1 missile |****** 551/N .... (8%) 2 missiles|*** 4215/N .... (59%) 3 missiles|************************ 1982/N .... (28%) 4 missiles|***** 450/N .... (6%)

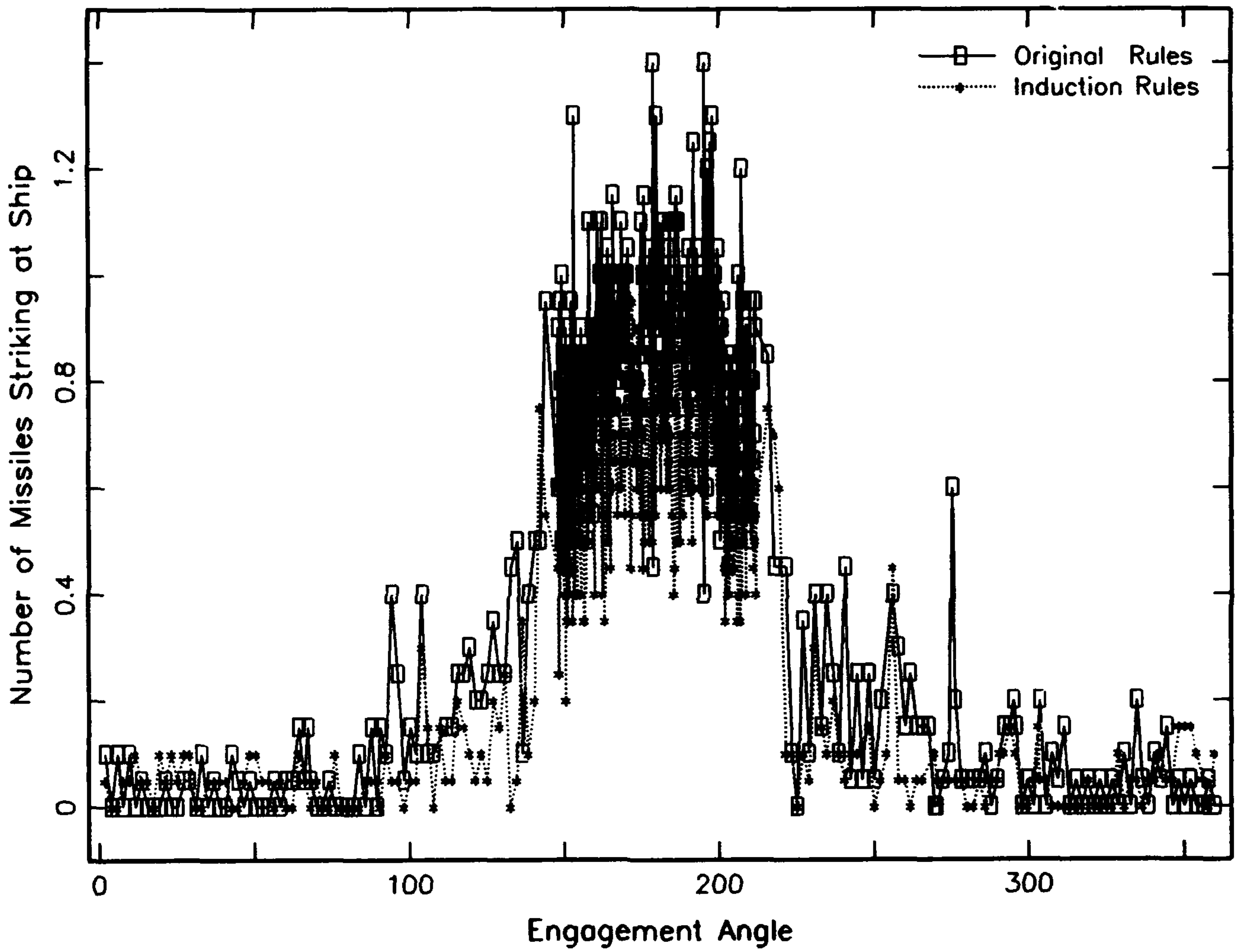

Figure 4-5 Comparison of Number of Missiles Striking Ship with Two Different Rules (1 fighter with 2 missiles)

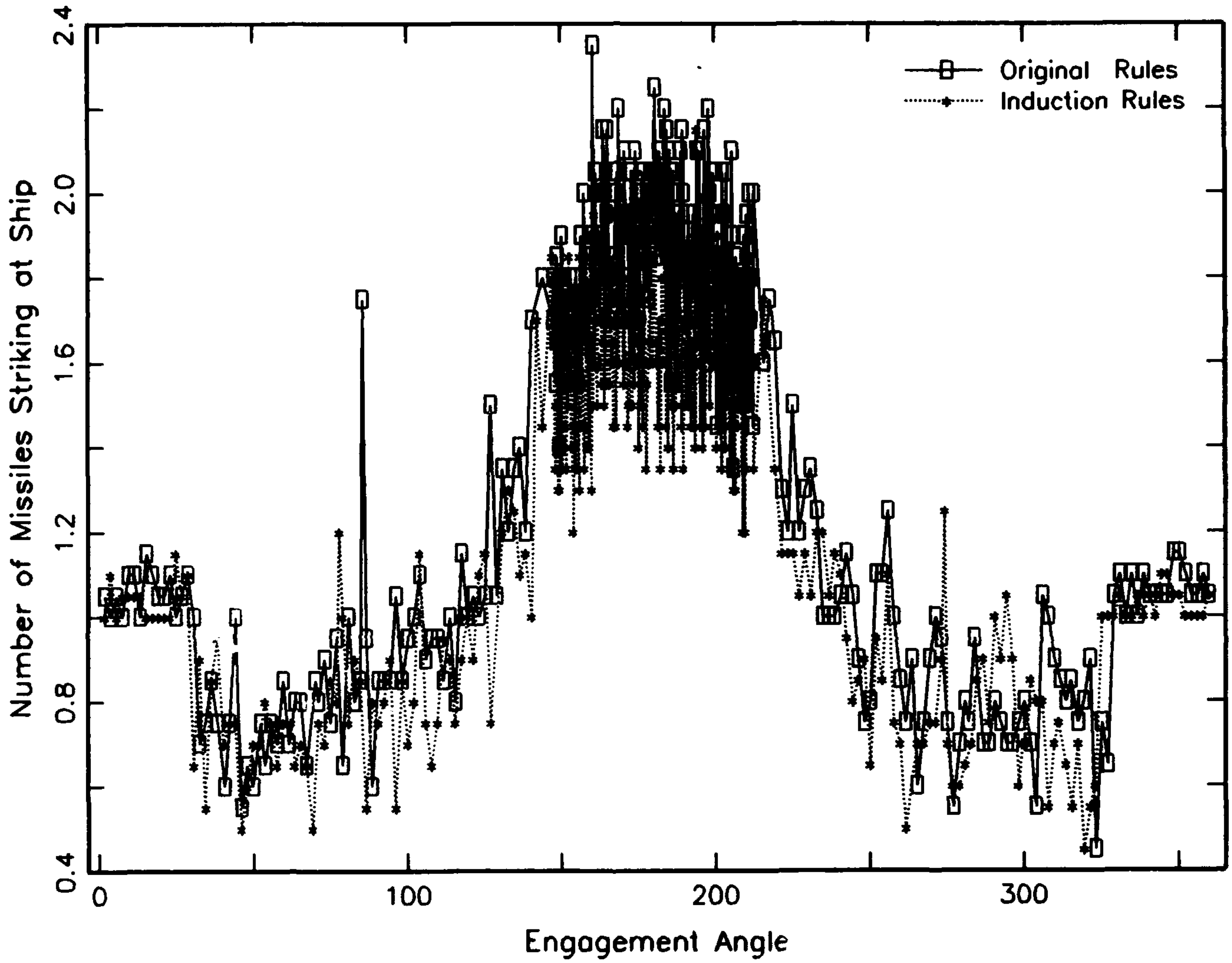

Figure 4-6 Comparison of Number of Missiles Striking Ship with Two Different Rules (1 fighter with 3 missiles)

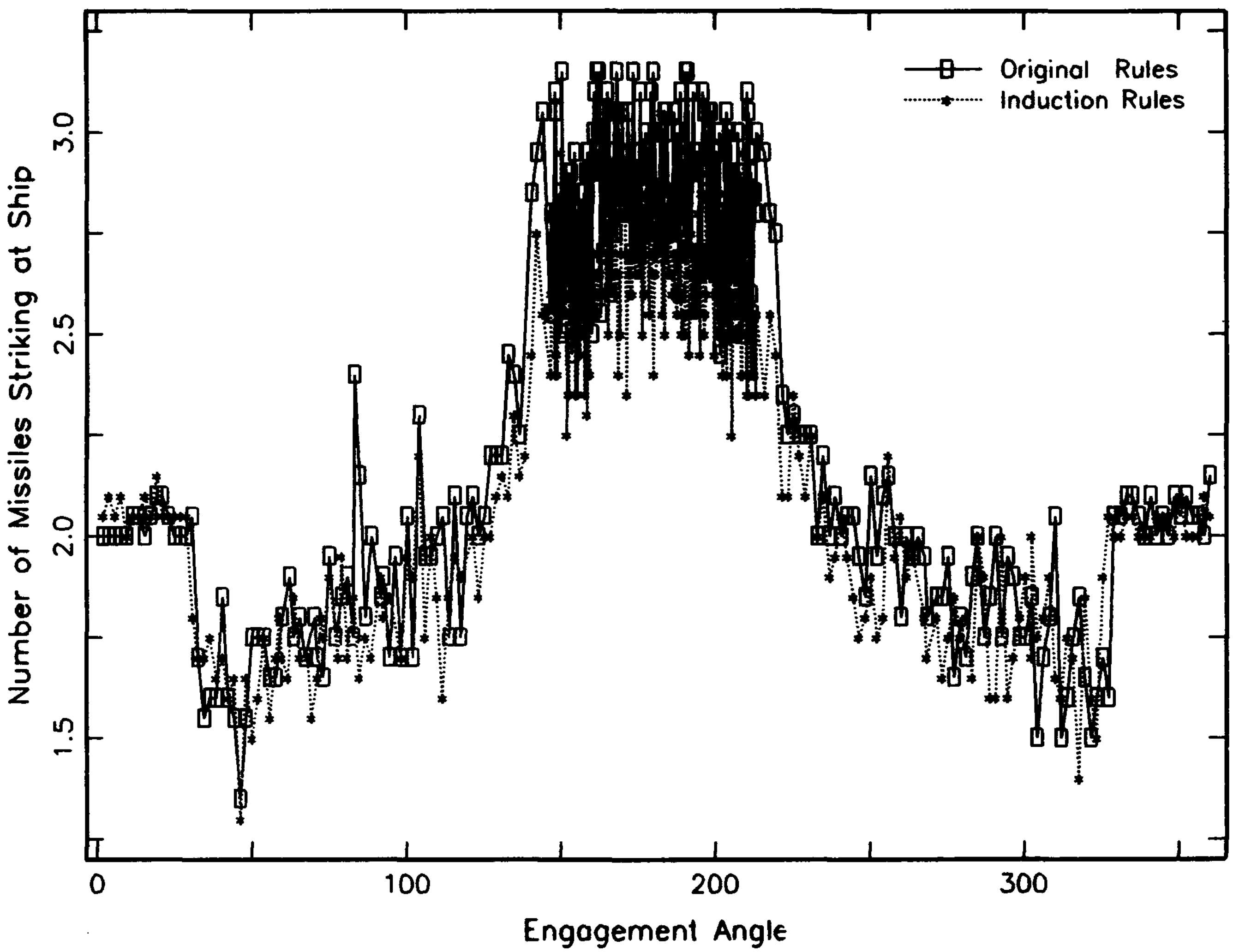

Figure 4-7 Comparison of Number of Missiles Striking Ship with Two Differen Rules (1 fighter with 4 missiles)

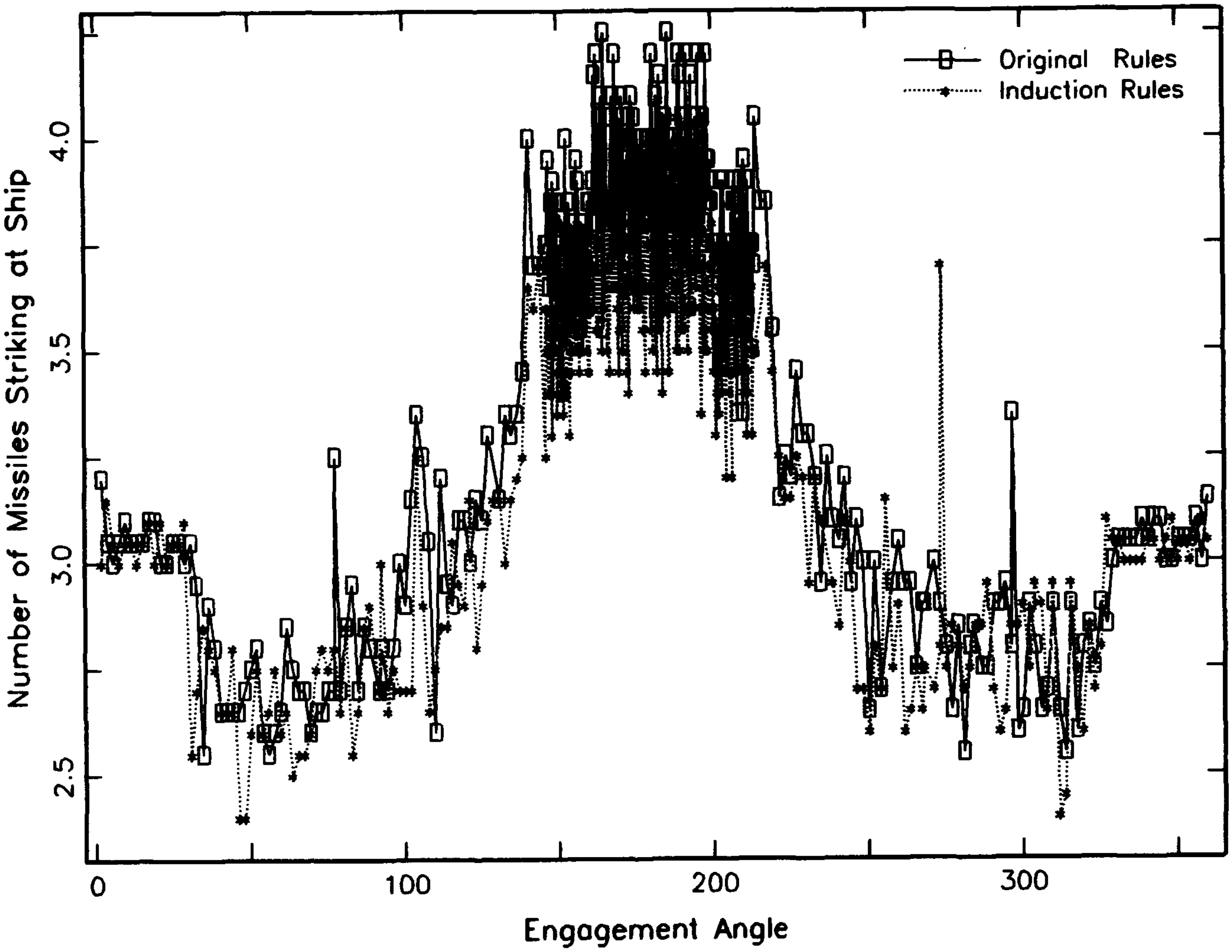

Figure 4-8 Comparison of Number of Missiles Striking Ship with Two Differen Rules (1 fighter with 5 missiles)

V. SUMMARY AND CONCLUSION

A. Summary

This thesis presents an effective method to integrate a simulation model with a powerful analytical tool (inductive learning) to analyze the ship air-defense combat model.

The simulation facilitates a top-down approach to modularized the interactions between ship and enemy targets. The user can easily modify the sub-model and input data to suit specific design requirement and evaluate alternative tactics.

The tasks accomplished in this thesis included:

1. Defining and setting up mission attributes for a simulation scenario to perform a particular mission function. (Section B and C of Chapter II)
2. Modularizing the simulation program and interactions between ship and targets (Section D, E and F of Chapter II) including:
   (1) The Structure Diagram of simulation,
   (2) Target Operation Process,
   (3) Ship Motion Control Process,
   (4) Detection, Threat Evaluation & Weapon Assign Process
   (5) Fire Control System Process,
   (6) Gun System Process,
3. Analyzing the general mechanism of inductive learning (Section A and B of Chapter III)
4. Detailing the ID3 algorithm for inductive learning (Section C of Chapter III)
5. Enhancing ID3 algorithm using χ^2 test and post-pruning decision tree (Section D of Chapter III.)
6. Implementing ID3 and χ^2 algorithms using POP-11, and linking simulation code in FORTRAN under POPLOG environment (Section E of Chapter III)
7. Data processing and analyses between simulation and inductive learning (Chapter IV)

By combining inductive rule generation and deductive simulation on ship air-defense, we showed that we could design better rules for weapon assignment. This can be seen from: (1) a rule with smaller size and more compact and precise operations for weapon assignment, (2) the decline of the number of missiles hitting the ship, and (3) the increase of the number of missiles destroyed by 5 in guns.

B. Conclusion

This research used a selected set of attributes and set of weapons for inductive learning and simulation. In the future, it is desirable to include all weapons available onboard and all possible attributes in the process. As the credibility of an induction rule depends on the accuracy of the simulation model, the impact of the missile striking at or detonating near the ship, in addition to weapon reliability or availability, should also be considered.

Currently, the results for simulation with rules generated by induction are compared with original rules using statistical methods (i.e., hypothesis testing). The issue of verification and validation of induction rules to increase their credibility should also be emphasized in future work.

The empirical evidence presented in this study showed that inductive learning can be an additional tool in war game analysis. It could aid the development of battle management strategies and find weakness or flaws in the ship air-defense system to improve AAW weapon effectiveness. Perhaps, simulation, inductive learning, and other analytical tools, such as regression analysis, can be integrated with exercises and analyses in a continuous cycle of research that allows each method to contribute its unique perspective.

APPENDIX A

DESCRIPTION OF MISSION ATTRIBUTES

Definition of Mission Attributes. The mission attributes of the ship air-defense evaluation model are defined. The attributes are discussed in order of presentation used by simulation program main menu for easy reference.

A. Detection, Threat Evaluation &Weapon Assign Random Variables

The attributes defined are for ranges of detection and reaction times used process 2 (Detection, Command and Control Process).

RNAGE SR DETECTION, (nm)

Search radar system detection range in nautical miles

ESM DET TO CS DETECT, (sec)

Reaction time from ESM detection until combat system detection occurs using search radar system

CS DET TO BEGIN TEWA, (sec)

Reaction time from combat system detection by search radar system until begin Threat Evaluation Weapon Assign

CS DET TO BEGIN VALID TRACK, (sec)

Reaction time from combat system detection by search radar system until begin valid track

LOSE VALID TRACK, (sec)

Reaction time from start of valid track of target suitable for ECS designation until lose valid track occurs

REGAIN VALID TRACK, (sec)

Reaction time from lose valid track until regain valid track of target

LOSE VALID TRK AGAIN, (sec)

Reaction time from regain valid track until subsequent lose valid track event

B. *Fire Control Random Variables*

The attributes defined are reaction times used in process 3 (Fire Control Precess).

DESIGN TO RDY, (sec)

Reaction time from FCS designation to a target until the ready to fire condition occurs

DESIGN TO ABORT, (sec)

Reaction time from FCS designation until abort when FCS fails to lockon to target

FCS RDY TO COAST, (sec)

Reaction time from FCS ready to fire until lose ready to fire and enter coast mode

FCS COAST TO EXIT, (sec)

Reaction time from FCS enter coast mode until time out occurs and FCS exit coast mode to either regain track or drop track

C. *Gun System Random Variables*

The attributes defined are reaction times used in process 4~5 and 6(Gun System Process)

ASSIGN TO GUN ABORT, (sec)

Reaction time from gun assign to target until gun ready to fire

ASSIGN TO GUN ABORT, (sec)

Reaction time from gun assign to target until abort due to failure to achieve gun synchronization with fire control

FIRE ONE ROUND, (sec)

Reaction time to load gun round and fire

KILL EVALUATION, (sec)

Reaction time to evaluate kill status of target at end of firing sequence when target killed or goes out of range

CLEAR GUN JAM, (sec)

Reaction time from gun ready to fire until lose gun synchronization with FCS

GUN RDY TO LOSE GUN RDY, (sec)

Reaction time from gun ready to fire until lose gun synchronization with FCS

LOSE GUN READY TO REGAIN, (sec)

Reaction time from lose gun ready until regain gun synchronization with FCS

D. Ship Probabilities

The attributes defined are probabilities of alternate outcomes in the six processes.

PROB GUN JAM

Probability of gun jam per round fired

PROB GUN HIT PER ROUND

Probability of target kill per round fired when target

PROB GUN HIT PER ROUND / JAMMING

Probability of target kill per round fired when target in jamming conditions

PROB FCS LOCKON

Probability FCS lock on the target

PROB LOSE FCS TRK

When FCS exits coast mode it must be decided if regain FCS track or drop track

PROB LOSE FCS TRK / JAMMING

When FCS exits coast mode in jamming condition it must be decided if regain FCS track or drop track

PROB GUN SYNCH (READY) AFTER DESIGNATION

When gun assigned to target it must be decided if gun synchronization with FCS will occur or must abort gun assign

PROB RELEASE GUN ON GNR

When gun loses synchronization with FCS it must be decided if gun to be released or wait for gun ready again.

APPENDIX B

MISSION ATTRIBUTE TABLE FOR SIMULATION SCENARIO

```
RNDV01 :(TGTYP 1)      1.    1.    1.    1.    0.    1.    0.
                       1.    1.    1.    1.    1.    1.    1.
                      70.   60.   30.    6.  120.   10.  150.
       (TGTYP 2)       1.    1.    1.    1.    0.    1.    0.
                       1.    1.    1.    1.    1.    1.    1.
                      70.   60.   30.    6.  120.   10.  150.
RNDV02 :(TGTYP 1)      1.    1.    1.    0.    1.
                       1.    1.    1.    1.    1.
                      15.   15.   25.  200.   20.
       (TGTYP 2)       1.    1.    1.    0.    1.
                       1.    1.    1.    1.    1.
                      15.   15.   25.  200.   20.
RNDV03 :(TGTYP 1)      1.    0.    1.    1.    0.    0.    0.
                       1.    1.    1.    1.    1.    1.    1.
                       5.   40.    4.    5.  240.  120.  180.
       (TGTYP 2)       1.    0.    1.    1.    0.    0.    0.
                       1.    1.    1.    1.    1.    1.    1.
                       5.   40.    4.    5.  240.  120.  180.
RNDV04 :(TGTYP 1)      1.    0.    1.    1.    0.    0.    0.
                       1.    1.    1.    1.    1.    1.    1.
                       5.   40.    4.    5.  240.  120.  180.
       (TGTYP 2)       1.    0.    1.    1.    0.    0.    0.
                       1.    1.    1.    1.    1.    1.    1.
                       5.   40.    4.    5.  240.  120.  180.
RNDV05 :(TGTYP 1)      1.    0.    1.    1.    0.    0.    0.
                       1.    1.    1.    1.    1.    1.    1.
                       5.   40.    4.    5.  240.  120.  180.
       (TGTYP 2)       1.    0.    1.    1.    0.    0.    0.
                       1.    1.    1.    1.    1.    1.    1.
                       5.   40.    4.    5.  240.  120.  180.
******************************** TARGET TYPE 1 ********************************
 P1   P2   P3   P4   P5   P6   P7   P8   P9  P10  P11  P12  P13  P14  P15  P16
 1.   1.   0.   0.3  0.   0.   1.   1.   0.   0.   0.   0.   0.   1.   1.   0.
************************************ cont. ************************************
 P17  P18  P19  P20  P21  P22  P23  HOME
 0.35 1.   1.   0.   0.35 1.   1.   1.
*********************************** TARGET TYPE 2 *****************************
 P1   P2   P3   P4   P5   P6   P7   P8   P9  P10  P11  P12  P13  P14  P15  P16
 1.   1.   0.   0.3  0.   0.   1.   1.   0.   0.   0.   0.   0.   1.   1.   0.
************************************ cont. ************************************
 P17  P18  P19  P20  P21  P22  P23  HOME
 0.35 1.   1.   0.   0.35 1.   1.   1.
****************************** WEAPON DESCRIPTIONS *****************************
FCS: 2G5N: 2G76: 1G40: 2 V1:0.98 V2:0.90 V3:0.95 V4:0.95
========================WPN1==WPN2==WPN3==WPN4==WPN5
  FCS OPEN  FIRE ANG :    0.    0.    0.    0.    0.
      CLOSE FIRE ANG :  360.  360.  360.  360.  360.
  5In. OPEN FIRE ANG :  220.   50.
      CLOSE FIRE ANG :  140.  290.
      MAGAZINE COUNT    700   700
  76mm OPEN FIRE ANG :  255.
      CLOSE FIRE ANG :  105.
      MAGAZINE COUNT :  700
  40mm OPEN FIRE ANG :   20.  200.
```

```
        CLOSE FIRE ANG :  160.  340.
        MAGAZINE COUNT : 2000  2000
    SHIP TARGET NUMBER :     5
******************************** TARGET DESCRIPTIONS *************************
         TYPE   BEGPRS  NXLNCH  Xcoor.   Ycoor.   Zcoor.   ROT.    TRNx    TRNy
target#1:  1      0.      2      0.     100.      0.     190.     0.      0.
target#2:  2     -1.      3      0.      27.      1.3    190.     0.      0.
target#3:  2     -1.      4      0.      27.      1.3    190.     0.1     0.
target#4:  2     -1.      5      0.      27.      1.3    190.     0.2     0.
target#4:  2     -1.      0      0.      27.      1.3    190.     0.3     0.
****************************** UNIT POSITION/VELOCITY ************************
     Xp: 0.0      Yp: 0.0      Zp: 0.0      Vx:20.0      Vy: 0.0      Vz: 0.0
********************************* TARGET PROGRAM *****************************
TYP1:  1.    7.    1.    1.   10.    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
     350.  400.  400.  400.  400.    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
       0.    0.    0.5   1.    1.    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
      32.   27.   26.   27.  100.    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
       1.3   1.3   1.3   1.3   1.3   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
TYP2:  4.    1.    6.   10.    0.    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
     650.  650.  600.  600.    0.    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
       0.    0.    0.    0.    0.    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
      26.   24.    2.7   0.    0.    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
       0.9   0.1   0.1   0.    0.    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
********************************* UNIT PROGRAM *******************************
TYP1:  3.    1.    5.    2.   12.    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
      71.   70.   20.  -70.  -90.    0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
TYP2:  3.    1.    5.    9.   11.   13.  15.  14.  12.  10.  16.   0.   0.   0.   0.
      71.   70.   20.    5.    4.    2.   0.  -1.  -3.  -4. -90.   0.   0.   0.   0.
```

APPENDIX C

ORIGINAL RULES OF WEAPON ASSIGNMENT

```
C
C      ***decision To Allocate Resources (Include Statements)
C
       if(ENTRY.ge.1.and.ENTRY.le.5) goto(510,599,530,540,550),ENTRY
C
C      ***FCS Resource
510          do 517 I=1,NSSFCS
             if(FCSANG(1,I).gt.FCSANG(2,I).and.
     +       (.NOT.(ANGLE.gt.FCSANG(1,I).and.
     +       ANGLE.le.360.or.ANGLE.ge.0.and.
     +       ANGLE.lt.FCSANG(2,I)))) goto 517
             if(FCSANG(1,I).le.FCSANG(2,I).and.
     +       (.NOT.(ANGLE.gt.FCSANG(1,I).and.
     +       ANGLE.lt.FCSANG(2,I)))) goto 517
             if(SURFCS(I).eq.1) goto 516
             if(SURWPN(TGTFCS(I)).ge.4) goto 517
             if(LPRIOR.eq.0) goto 515
             if(PRIOR(TGTFCS(I)).le.PRIOR(TGTFCS(LPRIOR))) goto 517
515          LPRIOR=I
             goto 517
516          NUMRES=I
             LOGIC=1
             return
517          continue
C           *Test For Pre-emption
             if(LPRIOR.eq.0) return
             if(PRIOR(NUMTGT).ge.PRIOR(TGTFCS(LPRIOR))) return
             NUMRES=LPRIOR
             PREMPT=1
             LOGIC=1
             return
C       ***G5N Resource
530          do 537 I=1,NUMG5N
             if(SURG5N(1,I).ne.1.or.G5NCNT(I).eq.0) goto 537
             if(G5NANG(1,I).gt.G5NANG(2,I).and.
     +       (.NOT.(ANGLE.gt.G5NANG(1,I).and.
     +       ANGLE.le.360.or.ANGLE.ge.0.and.
     +       ANGLE.lt.G5NANG(2,I)))) goto 537
             if(G5NANG(1,I).le.G5NANG(2,I).and.
     +       (.NOT.(ANGLE.gt.G5NANG(1,I).and.
     +       ANGLE.lt.G5NANG(2,I)))) goto 537
             if(SURG5N(1,I).eq.1.and.G5NCNT(I).gt.0) goto 536
             if(LPRIOR.eq.0) goto 535
             if(PRIOR(TGTG5N(I)).le.PRIOR(TGTG5N(LPRIOR))) goto 537
535          LPRIOR=I
             goto 537
536          NUMRES=I
             LOGIC=1
             return
537          continue
C           *test For Pre-emption
             if(LPRIOR.eq.0) return
             if(PRIOR(NUMTGT).ge.PRIOR(TGTG5N(LPRIOR))) return
             NUMRES=LPRIOR
             PREMPT=1
             LOGIC=1
             return
C       ***G76 Resource
```

```
540        do 547 I=1,NUMG76
           if(SURG76(1,I).ne.1.or.G76CNT(I).eq.0) goto 547
           if(G76ANG(1,I).gt.G76ANG(2,I).and.
     +     (.NOT.(ANGLE.gt.G76ANG(1,I).and.
     +     ANGLE.le.360.or.ANGLE.ge.0.and.
     +     ANGLE.lt.G76ANG(2,I)))) goto 547
           if(G76ANG(1,I).le.G76ANG(2,I).and.
     +     (.NOT.(ANGLE.gt.G76ANG(1,I).and.
     +     ANGLE.lt.G76ANG(2,I)))) goto 547
           if(SURG76(1,I).eq.1.and.G76CNT(I).gt.0) goto 546
           if(LPRIOR.eq.0) goto 545
           if(PRIOR(TGTG76(I)).le.PRIOR(TGTG76(LPRIOR))) goto 547
545        LPRIOR=I
           goto 547
546        NUMRES=I
           LOGIC=1
           return
547        continue
C         *Test For Pre-emption
           if(LPRIOR.eq.0) return
           if(PRIOR(NUMTGT).ge.PRIOR(TGTG76(LPRIOR))) return
           NUMRES=LPRIOR
           PREMPT=1
           LOGIC=1
           return
C     ***G40 Resource
550        do 557 I=1,NUMG40
           if(SURG40(1,I).ne.1.or.G40CNT(I).eq.0) goto 557
           if(G40ANG(1,I).gt.G40ANG(2,I).and.
     +     (.NOT.(ANGLE.gt.G40ANG(1,I).and.
     +     ANGLE.le.360.or.ANGLE.ge.0.and.
     +     ANGLE.lt.G40ANG(2,I)))) goto 557
           if(G40ANG(1,I).le.G40ANG(2,I).and.
     +     (.NOT.(ANGLE.gt.G40ANG(1,I).and.
     +     ANGLE.lt.G40ANG(2,I)))) goto 557
           if(SURG40(1,I).eq.1.and.G40CNT(I).gt.0) goto 556
           if(LPRIOR.eq.0) goto 555
           if(PRIOR(TGTG40(I)).le.PRIOR(TGTG40(LPRIOR))) goto 557
555        LPRIOR=I
           goto 557
556        NUMRES=I
           LOGIC=1
           return
557        continue
C         *test For Pre-emption
           if(LPRIOR.eq.0) return
           if(PRIOR(NUMTGT).ge.PRIOR(TGTG40(LPRIOR))) return
           NUMRES=LPRIOR
           PREMPT=1
           LOGIC=1
599        return
C
C***********************************************************************
```

APPENDIX D

LEARNING SAMPLES OF WEAPON ASSIGNMENT

```
;;;____________________5in
Guns Learning Sam-
ples____________________
_
[[LOGICAL
SURG5N_L_1__1_R_ S1 S2 S5]
 [LOGICAL SURG5N_L_1__2_R_
S1 S2 S5]
 [INTEGER ANGLE]
  [INTEGER PRIOR_L_NUMT-
GT_R_]
 [INTEGER PRIOR_L_TGTG5N_-
L_1_R__R_]
 [INTEGER PRIOR_L_TGTG5N_-
L_2_R__R_]
 ] -> Attribute_G5N;
[CLASSES G5N_ALLOCATE
     LOGIC___1   LOGIC___2
LOGIC___0] -> Class_G5N;
[
[ S1 S1 150 2 0 0  LOGIC___2]
[ S1 S2 150 2 0 2  LOGIC___0]
[ S1 S1 150 1 0 0  LOGIC___2]
[ S1 S1 149 2 0 0  LOGIC___2]
[ S1 S2 149 2 0 2  LOGIC___0]
[ S1 S1 149 1 0 0  LOGIC___2]
[ S1 S1 148 2 0 0  LOGIC___2]
[ S1 S2 148 2 0 2  LOGIC___0]
[ S1 S1 148 1 0 0  LOGIC___2]
[ S1 S1 147 2 0 0  LOGIC___2]
[ S1 S1 146 2 0 0  LOGIC___2]
[ S1 S1 145 2 0 0  LOGIC___2]
[ S1 S1 144 2 0 0  LOGIC___0]
[ S1 S2 147 2 0 2  LOGIC___0]
[ S1 S1 143 2 0 0  LOGIC___2]
[ S1 S1 142 2 0 0  LOGIC___2]
[ S1 S2 146 2 0 2  LOGIC___0]
[ S1 S2 145 2 0 2  LOGIC___0]
[ S1 S1 141 2 0 0  LOGIC___0]
[ S1 S2 144 2 0 2  LOGIC___0]
[ S1 S1 140 2 0 0  LOGIC___2]
[ S1 S2 143 2 0 2  LOGIC___0]
[ S2 S1 143 2 2 0  LOGIC___2]
[ S1 S1 139 2 0 0  LOGIC___1]
[ S2 S1 142 2 2 0  LOGIC___2]
[ S1 S1 138 2 0 0  LOGIC___0]
[ S2 S1 141 2 2 0  LOGIC___2]
[ S1 S1 137 2 0 0  LOGIC___1]
[ S2 S1 140 2 2 0  LOGIC___2]
[ S1 S1 147 1 0 0  LOGIC___2]
[ S1 S1 136 2 0 0  LOGIC___1]
[ S2 S1 139 2 2 0  LOGIC___2]
[ S1 S1 135 2 0 0  LOGIC___1]
[ S2 S1 138 2 2 0  LOGIC___2]
[ S2 S1 137 2 2 0  LOGIC___2]
[ S1 S1 134 2 0 0  LOGIC___1]
[ S1 S1 133 2 0 0  LOGIC___1]
[ S2 S1 136 2 2 0  LOGIC___2]
[ S1 S1 132 2 0 0  LOGIC___1]
[ S2 S1 135 2 2 0  LOGIC___0]
[ S2 S1 134 2 2 0  LOGIC___2]
[ S1 S1 131 2 0 0  LOGIC___1]
[ S2 S1 133 2 2 0  LOGIC___2]
[ S1 S1 130 2 0 0  LOGIC___1]
[ S2 S1 132 2 2 0  LOGIC___2]
[ S1 S1 129 2 0 0  LOGIC___1]
[ S2 S1 131 2 2 0  LOGIC___2]
[ S1 S1 128 2 0 0  LOGIC___1]
[ S2 S1 130 2 2 0  LOGIC___0]
[ S1 S1 127 2 0 0  LOGIC___1]
[ S2 S1 129 2 2 0  LOGIC___2]
[ S1 S1 125 2 0 0  LOGIC___1]
[ S2 S1 128 2 2 0  LOGIC___0]
[ S2 S1 127 2 2 0  LOGIC___2]
[ S1 S1 124 2 0 0  LOGIC___1]
[ S2 S1 126 2 2 0  LOGIC___2]
[ S1 S1 142 1 0 0  LOGIC___0]
[ S1 S1 123 2 0 0  LOGIC___1]
[ S2 S1 125 2 2 0  LOGIC___2]
[ S1 S1 122 2 0 0  LOGIC___1]
[ S2 S1 124 2 2 0  LOGIC___2]
[ S1 S1 121 2 0 0  LOGIC___1]
[ S2 S1 123 2 2 0  LOGIC___2]
[ S1 S1 120 2 0 0  LOGIC___1]
[ S2 S1 122 2 2 0  LOGIC___2]
[ S1 S1 119 2 0 0  LOGIC___1]
[ S2 S1 121 2 2 0  LOGIC___2]
[ S1 S1 118 2 0 0  LOGIC___1]
[ S2 S1 120 2 2 0  LOGIC___0]
[ S1 S1 137 1 0 0  LOGIC___1]
[ S1 S1 117 2 0 0  LOGIC___1]
[ S2 S1 119 2 2 0  LOGIC___2]
[ S1 S1 116 2 0 0  LOGIC___1]
[ S2 S1 118 2 2 0  LOGIC___2]
[ S2 S1 117 2 2 0  LOGIC___2]
[ S1 S1 115 2 0 0  LOGIC___1]
[ S1 S1 114 2 0 0  LOGIC___1]
[ S2 S1 116 2 2 0  LOGIC___0]
[ S1 S1 113 2 0 0  LOGIC___1]
[ S2 S1 115 2 2 0  LOGIC___2]
[ S2 S1 114 2 2 0  LOGIC___2]
[ S1 S1 112 2 0 0  LOGIC___0]
[ S2 S1 113 2 2 0  LOGIC___2]
[ S1 S1 111 2 0 0  LOGIC___1]
[ S1 S1 110 2 0 0  LOGIC___1]
[ S2 S1 112 2 2 0  LOGIC___2]
[ S2 S1 111 2 2 0  LOGIC___2]
[ S1 S1 109 2 0 0  LOGIC___1]
[ S2 S1 110 2 2 0  LOGIC___2]
[ S1 S1 108 2 0 0  LOGIC___1]
[ S2 S1 109 2 2 0  LOGIC___2]
[ S1 S1 107 2 0 0  LOGIC___1]
[ S2 S1 108 2 2 0  LOGIC___0]
[ S1 S1 143 1 0 0  LOGIC___0]
[ S1 S1 106 2 0 0  LOGIC___1]
[ S1 S1 124 1 0 0  LOGIC___1]
[ S2 S1 107 2 2 0  LOGIC___2]
[ S1 S1 105 2 0 0  LOGIC___1]
[ S1 S1 141 1 0 0  LOGIC___0]
[ S2 S1 106 2 2 0  LOGIC___2]
[ S1 S1 104 2 0 0  LOGIC___1]
[ S1 S1 103 2 0 0  LOGIC___1]
[ S2 S1 105 2 2 0  LOGIC___2]
[ S2 S1 104 2 2 0  LOGIC___2]
[ S1 S1 102 2 0 0  LOGIC___1]
[ S2 S1 103 2 2 0  LOGIC___2]
[ S1 S1 101 2 0 0  LOGIC___1]
[ S1 S1 119 1 0 0  LOGIC___1]
[ S2 S1 102 2 2 0  LOGIC___2]
[ S1 S1 100 2 0 0  LOGIC___1]
[ S2 S1 101 2 2 0  LOGIC___2]
[ S1 S1 99 2 0 0  LOGIC___1]
[ S1 S1 117 1 0 0  LOGIC___1]
[ S2 S1 100 2 2 0  LOGIC___2]
[ S1 S1  98 2 0 0  LOGIC___1]
[ S2 S1  99 2 2 0  LOGIC___2]
[ S1 S1  97 2 0 0  LOGIC___1]
[ S2 S1  98 2 2 0  LOGIC___2]
[ S1 S1  96 2 0 0  LOGIC___1]
[ S2 S1  97 2 2 0  LOGIC___0]
[ S1 S1 122 1 0 0  LOGIC___1]
[ S1 S1  95 2 0 0  LOGIC___1]
[ S2 S1  96 2 2 0  LOGIC___2]
[ S1 S1  94 2 0 0  LOGIC___1]
[ S2 S1  95 2 2 0  LOGIC___2]
[ S1 S1  93 2 0 0  LOGIC___1]
[ S2 S1  94 2 2 0  LOGIC___2]
[ S1 S1  92 2 0 0  LOGIC___1]
[ S2 S1  93 2 2 0  LOGIC___2]
[ S1 S1  91 2 0 0  LOGIC___1]
[ S2 S1  92 2 2 0  LOGIC___0]
[ S1 S1  90 2 0 0  LOGIC___1]
[ S2 S1  91 2 2 0  LOGIC___2]
[ S2 S1  89 2 2 0  LOGIC___2]
[ S2 S1  88 2 2 0  LOGIC___2]
[ S1 S1  89 2 0 0  LOGIC___1]
[ S1 S1  88 2 0 0  LOGIC___1]
[ S2 S1  87 2 2 0  LOGIC___0]
[ S1 S1 111 1 0 0  LOGIC___1]
[ S1 S1  87 2 0 0  LOGIC___1]
[ S2 S1  86 2 2 0  LOGIC___2]
[ S1 S1  86 2 0 0  LOGIC___1]
[ S2 S1  85 2 2 0  LOGIC___2]
[ S1 S1  85 2 0 0  LOGIC___1]
[ S2 S1  84 2 2 0  LOGIC___2]
[ S1 S1  98 1 0 0  LOGIC___1]
[ S1 S1  84 2 0 0  LOGIC___1]
[ S2 S1  83 2 2 0  LOGIC___2]
[ S1 S1  83 2 0 0  LOGIC___1]
[ S2 S1  82 2 2 0  LOGIC___2]
[ S1 S1  82 2 0 0  LOGIC___1]
```

[S2 S1 81 2 2 0 LOGIC___2]
[S1 S1 81 2 0 0 LOGIC___1]
[S2 S1 80 2 2 0 LOGIC___2]
[S1 S1 80 2 0 0 LOGIC___1]
[S2 S1 79 2 2 0 LOGIC___0]
[S1 S1 93 1 0 0 LOGIC___1]
[S1 S1 79 2 0 0 LOGIC___1]
[S2 S1 78 2 2 0 LOGIC___2]
[S1 S1 78 2 0 0 LOGIC___1]
[S2 S1 77 2 2 0 LOGIC___2]
[S1 S1 77 2 0 0 LOGIC___1]
[S2 S1 76 2 2 0 LOGIC___2]
[S1 S1 108 1 0 0 LOGIC___1]
[S1 S1 76 2 0 0 LOGIC___1]
[S2 S1 75 2 2 0 LOGIC___2]
[S1 S1 75 2 0 0 LOGIC___1]
[S2 S1 74 2 2 0 LOGIC___0]
[S1 S1 74 2 0 0 LOGIC___1]
[S2 S1 73 2 2 0 LOGIC___2]
[S1 S1 73 2 0 0 LOGIC___1]
[S2 S1 72 2 2 0 LOGIC___0]
[S1 S1 94 1 0 0 LOGIC___1]
[S1 S1 72 2 0 0 LOGIC___1]
[S2 S1 71 2 2 0 LOGIC___2]
[S1 S1 71 2 0 0 LOGIC___1]
[S2 S1 70 2 2 0 LOGIC___2]
[S1 S1 70 2 0 0 LOGIC___1]
[S2 S1 69 2 2 0 LOGIC___2]
[S1 S1 69 2 0 0 LOGIC___1]
[S2 S1 68 2 2 0 LOGIC___2]
[S1 S1 68 2 0 0 LOGIC___1]
[S1 S1 80 1 0 0 LOGIC___1]
[S2 S1 67 2 2 0 LOGIC___2]
[S1 S1 67 2 0 0 LOGIC___1]
[S2 S1 66 2 2 0 LOGIC___0]
[S1 S1 86 1 0 0 LOGIC___1]
[S1 S1 66 2 0 0 LOGIC___1]
[S2 S1 65 2 2 0 LOGIC___2]
[S1 S1 65 2 0 0 LOGIC___1]
[S1 S1 77 1 0 0 LOGIC___1]
[S2 S1 64 2 2 0 LOGIC___0]
[S1 S1 64 2 0 0 LOGIC___1]
[S2 S1 63 2 2 0 LOGIC___2]
[S1 S1 63 2 0 0 LOGIC___1]
[S2 S1 62 2 2 0 LOGIC___2]
[S1 S1 62 2 0 0 LOGIC___1]
[S2 S1 61 2 2 0 LOGIC___2]
[S1 S1 61 2 0 0 LOGIC___1]
[S2 S1 60 2 2 0 LOGIC___2]
[S1 S1 60 2 0 0 LOGIC___1]
[S2 S1 59 2 2 0 LOGIC___2]
[S1 S1 59 2 0 0 LOGIC___1]
[S2 S1 58 2 2 0 LOGIC___2]
[S1 S1 58 2 0 0 LOGIC___1]
[S2 S1 57 2 2 0 LOGIC___2]
[S1 S1 57 2 0 0 LOGIC___1]
[S2 S1 56 2 2 0 LOGIC___0]
[S1 S1 74 1 0 0 LOGIC___1]
[S1 S1 56 2 0 0 LOGIC___1]
[S2 S1 55 2 2 0 LOGIC___2]
[S1 S1 55 2 0 0 LOGIC___1]
[S2 S1 54 2 2 0 LOGIC___2]
[S1 S1 54 2 0 0 LOGIC___1]
[S2 S1 54 2 1 0 LOGIC___2]
[S2 S1 53 2 1 0 LOGIC___0]
[S1 S1 110 1 0 0 LOGIC___1]
[S1 S1 53 2 0 0 LOGIC___1]
[S2 S1 52 2 1 0 LOGIC___2]
[S1 S1 52 2 0 0 LOGIC___1]
[S2 S1 51 2 1 0 LOGIC___2]
[S1 S1 51 2 0 0 LOGIC___1]
[S2 S1 50 2 1 0 LOGIC___0]
[S1 S1 60 1 0 0 LOGIC___1]
[S1 S1 50 2 0 0 LOGIC___1]
[S1 S1 84 1 0 0 LOGIC___1]
[S2 S1 49 2 1 0 LOGIC___0]
[S1 S1 49 2 0 0 LOGIC___1]
[S2 S1 48 2 1 0 LOGIC___0]
[S1 S1 71 1 0 0 LOGIC___1]
[S1 S1 48 2 0 0 LOGIC___1]
[S2 S1 47 2 1 0 LOGIC___0]
[S1 S1 47 2 0 0 LOGIC___1]
[S1 S1 56 1 0 0 LOGIC___1]
[S1 S1 46 2 0 0 LOGIC___1]
[S2 S1 46 2 1 0 LOGIC___0]
[S1 S1 61 1 0 0 LOGIC___1]
[S2 S1 45 2 1 0 LOGIC___0]
[S1 S1 45 2 0 0 LOGIC___1]
[S2 S1 44 2 1 0 LOGIC___0]
[S1 S1 53 1 0 0 LOGIC___1]
[S1 S1 44 2 0 0 LOGIC___0]
[S2 S1 43 2 1 0 LOGIC___0]
[S1 S1 75 1 0 0 LOGIC___1]
[S1 S1 43 2 0 0 LOGIC___1]
[S1 S1 79 1 0 0 LOGIC___1]
[S2 S1 42 2 1 0 LOGIC___0]
[S1 S1 50 1 0 0 LOGIC___1]
[S1 S1 42 2 0 0 LOGIC___1]
[S1 S1 41 2 0 0 LOGIC___1]
[S2 S1 41 2 1 0 LOGIC___0]
[S1 S1 49 1 0 0 LOGIC___1]
[S2 S1 40 2 1 0 LOGIC___0]
[S1 S1 54 1 0 0 LOGIC___1]
[S1 S1 40 2 0 0 LOGIC___1]
[S1 S1 107 1 0 0 LOGIC___1]
[S2 S1 39 2 1 0 LOGIC___0]
[S1 S1 39 2 0 0 LOGIC___1]
[S1 S1 67 1 0 0 LOGIC___1]
[S1 S1 38 2 0 0 LOGIC___1]
[S2 S1 38 2 1 0 LOGIC___0]
[S1 S1 83 1 0 0 LOGIC___1]
[S1 S1 55 1 0 0 LOGIC___1]
[S1 S1 37 2 0 0 LOGIC___1]
[S2 S1 37 2 1 0 LOGIC___0]
[S1 S1 36 2 0 0 LOGIC___1]
[S2 S1 36 2 1 0 LOGIC___0]
[S1 S1 48 1 0 0 LOGIC___1]
[S2 S1 35 2 1 0 LOGIC___0]
[S1 S1 52 1 0 0 LOGIC___1]
[S1 S1 35 2 0 0 LOGIC___1]
[S1 S1 34 2 0 0 LOGIC___1]
[S2 S1 34 2 1 0 LOGIC___0]
[S2 S1 33 2 1 0 LOGIC___0]
[S1 S1 33 2 0 0 LOGIC___1]
[S1 S1 39 1 0 0 LOGIC___1]
[S1 S1 32 2 0 0 LOGIC___1]
[S2 S1 32 2 1 0 LOGIC___0]
[S1 S1 38 1 0 0 LOGIC___1]
[S1 S1 31 2 0 0 LOGIC___1]
[S2 S1 31 2 1 0 LOGIC___0]
[S1 S1 42 1 0 0 LOGIC___1]
[S1 S1 30 2 0 0 LOGIC___1]
[S2 S1 30 2 1 0 LOGIC___0]
[S1 S1 44 1 0 0 LOGIC___1]
[S2 S1 29 2 1 0 LOGIC___0]
[S1 S1 35 1 0 0 LOGIC___1]
[S1 S1 29 2 0 0 LOGIC___1]
[S2 S1 28 2 1 0 LOGIC___0]
[S1 S1 45 1 0 0 LOGIC___1]
[S1 S1 28 2 0 0 LOGIC___1]
[S1 S1 27 2 0 0 LOGIC___1]
[S2 S1 27 2 1 0 LOGIC___0]
[S1 S1 37 1 0 0 LOGIC___1]
[S1 S1 32 1 0 0 LOGIC___1]
[S1 S1 26 2 0 0 LOGIC___1]
[S2 S1 26 2 1 0 LOGIC___0]
[S1 S1 31 1 0 0 LOGIC___1]
[S2 S1 25 2 1 0 LOGIC___0]
[S1 S1 25 2 0 0 LOGIC___1]
[S1 S1 51 1 0 0 LOGIC___1]
[S1 S1 24 2 0 0 LOGIC___1]
[S2 S1 24 2 1 0 LOGIC___0]
[S1 S1 33 1 0 0 LOGIC___1]
[S1 S1 23 2 0 0 LOGIC___1]
[S2 S1 23 2 1 0 LOGIC___0]
[S2 S1 22 2 1 0 LOGIC___0]
[S1 S1 27 1 0 0 LOGIC___1]
[S1 S1 22 2 0 0 LOGIC___1]
[S2 S1 21 2 1 0 LOGIC___0]
[S1 S1 21 2 0 0 LOGIC___1]
[S1 S1 28 1 0 0 LOGIC___1]
[S1 S1 20 2 0 0 LOGIC___1]
[S2 S1 20 2 1 0 LOGIC___0]
[S1 S1 24 1 0 0 LOGIC___1]
[S1 S1 19 2 0 0 LOGIC___1]
[S2 S1 19 2 1 0 LOGIC___0]
[S2 S1 18 2 1 0 LOGIC___0]
[S1 S1 22 1 0 0 LOGIC___1]
[S1 S1 18 2 0 0 LOGIC___1]
[S1 S1 21 1 0 0 LOGIC___1]
[S1 S1 17 2 0 0 LOGIC___1]
[S2 S1 17 2 1 0 LOGIC___0]
[S1 S1 20 1 0 0 LOGIC___1]
[S1 S1 16 2 0 0 LOGIC___1]
[S2 S1 16 2 1 0 LOGIC___0]
[S1 S1 19 1 0 0 LOGIC___1]
[S1 S1 15 2 0 0 LOGIC___1]
[S2 S1 15 2 1 0 LOGIC___0]
[S1 S1 18 1 0 0 LOGIC___1]
[S2 S1 14 2 1 0 LOGIC___0]
[S1 S1 14 2 0 0 LOGIC___1]
[S1 S1 13 2 0 0 LOGIC___1]
[S2 S1 13 2 1 0 LOGIC___0]
[S1 S1 34 1 0 0 LOGIC___1]
[S1 S1 15 1 0 0 LOGIC___1]
[S1 S1 12 2 0 0 LOGIC___1]
[S2 S1 12 2 1 0 LOGIC___0]
[S1 S1 14 1 0 0 LOGIC___1]
[S1 S1 11 2 0 0 LOGIC___1]
[S2 S1 11 2 1 0 LOGIC___0]
[S1 S1 10 2 0 0 LOGIC___1]
[S2 S1 10 2 1 0 LOGIC___0]

[S1 S1 35 1 1 1 LOGIC___1]
[S1 S1 12 1 0 0 LOGIC___1]
[S1 S1 9 2 0 0 LOGIC___1]
[S2 S1 9 2 1 0 LOGIC___0]
[S1 S1 32 1 1 1 LOGIC___1]
[S1 S1 8 2 0 0 LOGIC___1]
[S2 S1 8 2 1 0 LOGIC___0]
[S1 S1 10 1 0 0 LOGIC___1]
[S2 S1 7 2 1 0 LOGIC___0]
[S1 S1 11 1 0 0 LOGIC___1]
[S1 S1 7 2 0 0 LOGIC___1]
[S1 S1 6 2 0 0 LOGIC___1]
[S2 S1 6 2 1 0 LOGIC___0]
[S1 S1 17 1 1 1 LOGIC___1]
[S1 S1 13 1 1 1 LOGIC___1]
[S1 S1 5 2 0 0 LOGIC___1]
[S2 S1 5 2 1 0 LOGIC___0]
[S1 S1 6 1 0 0 LOGIC___1]
[S1 S1 7 1 0 0 LOGIC___1]
[S1 S1 4 2 0 0 LOGIC___1]
[S2 S1 4 2 1 0 LOGIC___0]
[S1 S1 22 1 1 1 LOGIC___1]
[S1 S1 3 2 0 0 LOGIC___1]
[S2 S1 3 2 1 0 LOGIC___0]
[S1 S1 19 1 1 1 LOGIC___1]
[S1 S1 4 1 0 0 LOGIC___1]
[S1 S1 2 2 0 0 LOGIC___1]
[S2 S1 2 2 1 0 LOGIC___0]
[S1 S1 3 1 0 0 LOGIC___1]
[S1 S1 2 1 0 0 LOGIC___1]
[S1 S1 1 2 0 0 LOGIC___1]
[S2 S1 1 2 1 0 LOGIC___0]
[S1 S1 2 1 1 1 LOGIC___1]
[S1 S1 1 1 1 1 LOGIC___1]
[S1 S1 360 2 0 0 LOGIC___1]
[S2 S1 360 2 1 0 LOGIC___0]
[S1 S1 360 1 0 0 LOGIC___1]
[S1 S1 359 2 0 0 LOGIC___1]
[S2 S1 359 2 1 0 LOGIC___0]
[S1 S1 359 1 0 0 LOGIC___1]
[S1 S1 358 2 0 0 LOGIC___1]
[S2 S1 358 2 1 0 LOGIC___0]
[S1 S1 358 1 0 0 LOGIC___1]
[S1 S1 357 1 0 0 LOGIC___1]
[S1 S1 357 2 0 0 LOGIC___1]
[S2 S1 357 2 1 0 LOGIC___0]
[S1 S1 209 1 1 1 LOGIC___2]
[S1 S1 356 2 0 0 LOGIC___1]
[S1 S1 355 1 0 0 LOGIC___1]
[S2 S1 356 2 1 0 LOGIC___0]
[S1 S1 351 1 1 1 LOGIC___1]
[S1 S1 355 2 0 0 LOGIC___1]
[S2 S1 355 2 1 0 LOGIC___0]
[S1 S1 353 1 0 0 LOGIC___1]
[S1 S1 354 2 0 0 LOGIC___1]
[S2 S1 354 2 1 0 LOGIC___0]
[S1 S1 347 1 1 1 LOGIC___1]
[S1 S1 352 1 0 0 LOGIC___1]
[S1 S1 353 2 0 0 LOGIC___1]
[S2 S1 353 2 1 0 LOGIC___0]
[S1 S1 351 1 0 0 LOGIC___1]
[S1 S1 352 2 0 0 LOGIC___1]
[S2 S1 352 2 1 0 LOGIC___0]
[S1 S1 348 1 0 0 LOGIC___1]
[S1 S1 185 1 1 1 LOGIC___0]
[S1 S1 351 2 0 0 LOGIC___1]
[S2 S1 351 2 1 0 LOGIC___0]
[S1 S1 347 1 0 0 LOGIC___1]
[S1 S1 349 1 0 0 LOGIC___1]
[S1 S1 350 2 0 0 LOGIC___1]
[S2 S1 350 2 1 0 LOGIC___0]
[S1 S1 346 1 0 0 LOGIC___1]
[S1 S1 349 2 0 0 LOGIC___1]
[S2 S1 349 2 1 0 LOGIC___0]
[S1 S1 344 1 0 0 LOGIC___1]
[S1 S1 348 2 0 0 LOGIC___1]
[S2 S1 348 2 1 0 LOGIC___0]
[S1 S1 347 2 0 0 LOGIC___1]
[S2 S1 347 2 1 0 LOGIC___0]
[S1 S1 341 1 0 0 LOGIC___1]
[S1 S1 346 2 0 0 LOGIC___1]
[S2 S1 346 2 1 0 LOGIC___0]
[S1 S1 325 1 0 0 LOGIC___1]
[S1 S1 345 2 0 0 LOGIC___1]
[S1 S1 340 1 0 0 LOGIC___1]
[S2 S1 345 2 1 0 LOGIC___0]
[S1 S1 296 1 1 1 LOGIC___1]
[S1 S1 344 2 0 0 LOGIC___1]
[S2 S1 344 2 1 0 LOGIC___0]
[S1 S1 332 1 0 0 LOGIC___1]
[S1 S1 241 1 0 0 LOGIC___1]
[S1 S1 343 2 0 0 LOGIC___1]
[S2 S1 343 2 1 0 LOGIC___0]
[S1 S1 329 1 0 0 LOGIC___1]
[S1 S1 342 2 0 0 LOGIC___1]
[S2 S1 342 2 1 0 LOGIC___0]
[S1 S1 289 1 0 0 LOGIC___1]
[S1 S1 341 2 0 0 LOGIC___1]
[S1 S1 330 1 0 0 LOGIC___1]
[S2 S1 341 2 1 0 LOGIC___0]
[S1 S1 334 1 0 0 LOGIC___1]
[S1 S1 340 2 0 0 LOGIC___1]
[S2 S1 340 2 1 0 LOGIC___0]
[S1 S1 336 1 0 0 LOGIC___1]
[S1 S1 284 1 0 0 LOGIC___1]
[S1 S1 339 2 0 0 LOGIC___1]
[S2 S1 339 2 1 0 LOGIC___0]
[S1 S1 335 1 0 0 LOGIC___1]
[S1 S1 338 2 0 0 LOGIC___1]
[S1 S1 321 1 0 0 LOGIC___1]
[S2 S1 338 2 1 0 LOGIC___0]
[S1 S1 296 1 0 0 LOGIC___1]
[S1 S1 337 2 0 0 LOGIC___1]
[S2 S1 337 2 1 0 LOGIC___0]
[S1 S1 333 1 0 0 LOGIC___1]
[S1 S1 336 2 0 0 LOGIC___1]
[S2 S1 336 2 1 0 LOGIC___0]
[S1 S1 322 1 0 0 LOGIC___1]
[S1 S1 324 1 0 0 LOGIC___1]
[S1 S1 335 2 0 0 LOGIC___1]
[S2 S1 335 2 1 0 LOGIC___0]
[S1 S1 323 1 0 0 LOGIC___1]
[S1 S1 334 2 0 0 LOGIC___1]
[S2 S1 334 2 1 0 LOGIC___0]
[S1 S1 333 2 0 0 LOGIC___1]
[S2 S1 333 2 1 0 LOGIC___0]
[S1 S1 320 1 0 0 LOGIC___1]
[S1 S1 332 2 0 0 LOGIC___1]
[S2 S1 332 2 1 0 LOGIC___0]
[S1 S1 316 1 0 0 LOGIC___1]
[S1 S1 331 2 0 0 LOGIC___1]
[S2 S1 331 2 1 0 LOGIC___0]
[S1 S1 330 2 0 0 LOGIC___1]
[S1 S1 317 1 0 0 LOGIC___1]
[S2 S1 330 2 1 0 LOGIC___0]
[S1 S1 329 2 0 0 LOGIC___1]
[S2 S1 329 2 1 0 LOGIC___0]
[S1 S1 319 1 0 0 LOGIC___1]
[S1 S1 279 1 0 0 LOGIC___1]
[S1 S1 328 2 0 0 LOGIC___1]
[S2 S1 328 2 1 0 LOGIC___0]
[S1 S1 313 1 0 0 LOGIC___1]
[S1 S1 327 2 0 0 LOGIC___1]
[S1 S1 303 1 0 0 LOGIC___1]
[S2 S1 327 2 1 0 LOGIC___0]
[S1 S1 315 1 0 0 LOGIC___1]
[S1 S1 326 2 0 0 LOGIC___1]
[S1 S1 275 1 0 0 LOGIC___1]
[S2 S1 326 2 1 0 LOGIC___0]
[S1 S1 300 1 0 0 LOGIC___1]
[S1 S1 325 2 0 0 LOGIC___1]
[S1 S1 324 2 0 0 LOGIC___1]
[S2 S1 325 2 1 0 LOGIC___0]
[S1 S1 312 1 0 0 LOGIC___1]
[S2 S1 324 2 1 0 LOGIC___0]
[S1 S1 323 2 0 0 LOGIC___1]
[S2 S1 323 2 1 0 LOGIC___0]
[S1 S1 322 2 0 0 LOGIC___1]
[S2 S1 322 2 1 0 LOGIC___0]
[S1 S1 309 1 0 0 LOGIC___1]
[S1 S1 321 2 0 0 LOGIC___1]
[S1 S1 304 1 0 0 LOGIC___1]
[S2 S1 321 2 1 0 LOGIC___0]
[S1 S1 314 1 0 0 LOGIC___1]
[S1 S1 320 2 0 0 LOGIC___1]
[S1 S1 307 1 0 0 LOGIC___1]
[S2 S1 320 2 1 0 LOGIC___0]
[S1 S1 302 1 0 0 LOGIC___1]
[S1 S1 319 2 0 0 LOGIC___1]
[S1 S1 301 1 0 0 LOGIC___1]
[S2 S1 319 2 1 0 LOGIC___0]
[S1 S1 318 2 0 0 LOGIC___1]
[S2 S1 318 2 1 0 LOGIC___0]
[S1 S1 317 2 0 0 LOGIC___1]
[S1 S1 298 1 0 0 LOGIC___1]
[S2 S1 317 2 1 0 LOGIC___0]
[S1 S1 316 2 0 0 LOGIC___1]
[S1 S1 308 1 0 0 LOGIC___1]
[S2 S1 316 2 1 0 LOGIC___0]
[S1 S1 315 2 0 0 LOGIC___1]
[S1 S1 314 2 0 0 LOGIC___1]
[S2 S1 315 2 1 0 LOGIC___0]
[S1 S1 294 1 0 0 LOGIC___1]
[S2 S1 314 2 1 0 LOGIC___0]
[S1 S1 299 1 0 0 LOGIC___1]
[S1 S1 313 2 0 0 LOGIC___1]
[S1 S1 281 1 0 0 LOGIC___1]
[S2 S1 313 2 1 0 LOGIC___0]
[S1 S1 280 1 0 0 LOGIC___1]
[S1 S1 312 2 0 0 LOGIC___1]
[S1 S1 287 1 0 0 LOGIC___1]
[S2 S1 312 2 1 0 LOGIC___0]

```
[S1 S1 311 2 0 0 LOGIC___1]
[S2 S1 311 2 1 0 LOGIC___0]
[S1 S1 278 1 0 0 LOGIC___1]
[S1 S1 310 2 0 0 LOGIC___1]
[S1 S1 285 1 0 0 LOGIC___1]
[S2 S1 310 2 1 0 LOGIC___0]
[S1 S1 271 1 0 0 LOGIC___1]
[S1 S1 309 2 0 0 LOGIC___1]
[S2 S1 309 2 1 0 LOGIC___0]
[S1 S1 252 1 0 0 LOGIC___1]
[S1 S1 308 2 0 0 LOGIC___1]
[S1 S1 273 1 0 0 LOGIC___1]
[S1 S1 307 2 0 0 LOGIC___1]
[S2 S1 308 2 1 0 LOGIC___0]
[S1 S1 306 2 0 0 LOGIC___1]
[S2 S1 307 2 1 0 LOGIC___0]
[S1 S1 283 1 0 0 LOGIC___1]
[S2 S1 306 2 1 0 LOGIC___0]
[S1 S1 282 1 0 0 LOGIC___1]
[S1 S1 305 2 0 0 LOGIC___1]
[S2 S1 306 2 2 0 LOGIC___0]
[S1 S1 231 1 0 0 LOGIC___1]
[S2 S1 305 2 2 0 LOGIC___0]
[S1 S1 295 1 0 0 LOGIC___1]
[S1 S1 304 2 0 0 LOGIC___1]
[S1 S1 303 2 0 0 LOGIC___1]
[S2 S1 304 2 2 0 LOGIC___0]
[S1 S1 293 1 0 0 LOGIC___1]
[S2 S1 303 2 2 0 LOGIC___0]
[S1 S1 302 2 0 0 LOGIC___1]
[S2 S1 302 2 2 0 LOGIC___0]
[S1 S1 208 1 0 0 LOGIC___2]
[S1 S1 301 2 0 0 LOGIC___1]
[S1 S1 260 1 0 0 LOGIC___1]
[S1 S1 300 2 0 0 LOGIC___1]
[S2 S1 301 2 2 0 LOGIC___0]
[S1 S1 299 2 0 0 LOGIC___1]
[S2 S1 300 2 2 0 LOGIC___0]
[S2 S1 299 2 2 0 LOGIC___0]
[S1 S1 272 1 0 0 LOGIC___1]
[S1 S1 298 2 0 0 LOGIC___1]
[S2 S1 298 2 2 0 LOGIC___0]
[S1 S1 240 1 0 0 LOGIC___1]
[S1 S1 297 2 0 0 LOGIC___1]
[S1 S1 286 1 0 0 LOGIC___1]
[S1 S1 296 2 0 0 LOGIC___1]
[S2 S1 297 2 2 0 LOGIC___0]
[S1 S1 227 2 0 0 LOGIC___1]
[S2 S1 296 2 2 0 LOGIC___0]
[S1 S1 239 1 0 0 LOGIC___1]
[S1 S1 295 2 0 0 LOGIC___1]
[S1 S1 267 1 0 0 LOGIC___1]
[S2 S1 295 2 2 0 LOGIC___0]
[S1 S1 254 1 0 0 LOGIC___1]
[S1 S1 294 2 0 0 LOGIC___1]
[S1 S1 274 1 0 0 LOGIC___1]
[S1 S1 293 2 0 0 LOGIC___1]
[S2 S1 294 2 2 0 LOGIC___0]
[S1 S1 265 1 0 0 LOGIC___1]
[S1 S1 263 1 0 0 LOGIC___1]
[S1 S1 292 2 0 0 LOGIC___1]
[S2 S1 293 2 2 0 LOGIC___0]
[S1 S1 262 1 0 0 LOGIC___1]
[S2 S1 292 2 2 0 LOGIC___0]
[S1 S1 291 2 0 0 LOGIC___1]
[S1 S1 290 2 0 0 LOGIC___1]
[S2 S1 291 2 2 0 LOGIC___0]
[S1 S1 248 1 0 0 LOGIC___1]
[S1 S1 289 2 0 0 LOGIC___1]
[S2 S1 290 2 2 0 LOGIC___0]
[S1 S1 246 1 0 0 LOGIC___1]
[S1 S1 288 2 0 0 LOGIC___1]
[S2 S1 289 2 2 0 LOGIC___0]
[S1 S1 268 1 0 0 LOGIC___1]
[S2 S1 288 2 2 0 LOGIC___2]
[S1 S1 287 2 0 0 LOGIC___1]
[S1 S1 286 2 0 0 LOGIC___1]
[S2 S1 287 2 2 0 LOGIC___2]
[S1 S1 285 2 0 0 LOGIC___1]
[S2 S1 285 2 2 0 LOGIC___2]
[S1 S1 284 2 0 0 LOGIC___1]
[S1 S1 283 2 0 0 LOGIC___1]
[S2 S1 284 2 2 0 LOGIC___2]
[S1 S1 282 2 0 0 LOGIC___1]
[S2 S1 283 2 2 0 LOGIC___2]
[S1 S1 281 2 0 0 LOGIC___1]
[S2 S1 282 2 2 0 LOGIC___2]
[S1 S1 280 2 0 0 LOGIC___1]
[S2 S1 281 2 2 0 LOGIC___2]
[S1 S1 279 2 0 0 LOGIC___1]
[S2 S1 280 2 2 0 LOGIC___2]
[S1 S1 278 2 0 0 LOGIC___1]
[S2 S1 279 2 2 0 LOGIC___2]
[S1 S1 233 1 0 0 LOGIC___1]
[S1 S1 277 2 0 0 LOGIC___1]
[S2 S1 278 2 2 0 LOGIC___2]
[S1 S1 276 2 0 0 LOGIC___1]
[S2 S1 277 2 2 0 LOGIC___2]
[S1 S1 275 2 0 0 LOGIC___1]
[S2 S1 276 2 2 0 LOGIC___0]
[S1 S1 274 2 0 0 LOGIC___1]
[S2 S1 275 2 2 0 LOGIC___2]
[S1 S1 273 2 0 0 LOGIC___1]
[S2 S1 274 2 2 0 LOGIC___2]
[S1 S1 272 2 0 0 LOGIC___1]
[S2 S1 273 2 2 0 LOGIC___2]
[S1 S1 238 1 0 0 LOGIC___1]
[S1 S1 271 2 0 0 LOGIC___1]
[S2 S1 272 2 2 0 LOGIC___0]
[S1 S1 270 2 0 0 LOGIC___1]
[S2 S1 271 2 2 0 LOGIC___2]
[S2 S1 269 2 2 0 LOGIC___2]
[S1 S1 269 2 0 0 LOGIC___1]
[S2 S1 268 2 2 0 LOGIC___0]
[S1 S1 268 2 0 0 LOGIC___1]
[S2 S1 267 2 2 0 LOGIC___2]
[S2 S1 266 2 2 0 LOGIC___2]
[S1 S1 267 2 0 0 LOGIC___1]
[S1 S1 266 2 0 0 LOGIC___1]
[S2 S1 265 2 2 0 LOGIC___2]
[S1 S1 265 2 0 0 LOGIC___1]
[S2 S1 264 2 2 0 LOGIC___0]
[S1 S1 213 1 0 0 LOGIC___0]
[S2 S1 263 2 2 0 LOGIC___2]
[S1 S1 264 2 0 0 LOGIC___1]
[S1 S1 263 2 0 0 LOGIC___1]
[S2 S1 262 2 2 0 LOGIC___0]
[S1 S1 217 3 0 0 LOGIC___0]
[S1 S1 262 2 0 0 LOGIC___1]
[S2 S1 261 2 2 0 LOGIC___2]
[S2 S1 260 2 2 0 LOGIC___2]
[S1 S1 260 2 0 0 LOGIC___1]
[S1 S1 261 2 0 0 LOGIC___1]
[S2 S1 259 2 2 0 LOGIC___0]
[S1 S1 243 1 0 0 LOGIC___1]
[S1 S1 259 2 0 0 LOGIC___1]
[S2 S1 258 2 2 0 LOGIC___2]
[S2 S1 257 2 2 0 LOGIC___2]
[S1 S1 258 2 0 0 LOGIC___1]
[S2 S1 256 2 2 0 LOGIC___2]
[S1 S1 257 2 0 0 LOGIC___1]
[S2 S1 255 2 2 0 LOGIC___2]
[S1 S1 256 2 0 0 LOGIC___1]
[S2 S1 254 2 2 0 LOGIC___2]
[S1 S1 255 2 0 0 LOGIC___1]
[S2 S1 253 2 2 0 LOGIC___2]
[S1 S1 254 2 0 0 LOGIC___1]
[S2 S1 252 2 2 0 LOGIC___2]
[S1 S1 253 2 0 0 LOGIC___1]
[S2 S1 251 2 2 0 LOGIC___2]
[S1 S1 252 2 0 0 LOGIC___1]
[S2 S1 250 2 2 0 LOGIC___0]
[S1 S1 234 1 0 0 LOGIC___1]
[S1 S1 251 2 0 0 LOGIC___1]
[S2 S1 249 2 2 0 LOGIC___2]
[S1 S1 250 2 0 0 LOGIC___1]
[S1 S1 223 1 0 0 LOGIC___1]
[S2 S1 248 2 2 0 LOGIC___2]
[S1 S1 249 2 0 0 LOGIC___1]
[S2 S1 247 2 2 0 LOGIC___2]
[S1 S1 247 2 0 0 LOGIC___1]
[S1 S1 248 2 0 0 LOGIC___1]
[S2 S1 246 2 2 0 LOGIC___2]
[S2 S1 245 2 2 0 LOGIC___0]
[S1 S1 229 1 0 0 LOGIC___1]
[S1 S1 246 2 0 0 LOGIC___1]
[S2 S1 244 2 2 0 LOGIC___2]
[S1 S1 245 2 0 0 LOGIC___1]
[S2 S1 243 2 2 0 LOGIC___2]
[S1 S1 244 2 0 0 LOGIC___1]
[S2 S1 242 2 2 0 LOGIC___2]
[S1 S1 243 2 0 0 LOGIC___1]
[S2 S1 241 2 2 0 LOGIC___2]
[S1 S1 241 2 0 0 LOGIC___1]
[S1 S1 242 2 0 0 LOGIC___1]
[S2 S1 240 2 2 0 LOGIC___2]
[S2 S1 239 2 2 0 LOGIC___0]
[S1 S1 240 2 0 0 LOGIC___1]
[S2 S1 238 2 2 0 LOGIC___2]
[S1 S1 238 2 0 0 LOGIC___1]
[S1 S1 239 2 0 0 LOGIC___1]
[S2 S1 237 2 2 0 LOGIC___2]
[S2 S1 236 2 2 0 LOGIC___2]
[S2 S1 235 2 2 0 LOGIC___2]
[S1 S1 237 2 0 0 LOGIC___1]
[S1 S1 219 1 0 0 LOGIC___0]
[S1 S1 236 2 0 0 LOGIC___1]
[S2 S1 234 2 2 0 LOGIC___2]
[S1 S1 235 2 0 0 LOGIC___1]
[S2 S1 233 2 2 0 LOGIC___0]
[S2 S1 232 2 2 0 LOGIC___2]
[S1 S1 234 2 0 0 LOGIC___1]
```

```
[ S1 S1 233 2 0 0 LOGIC___1]
[ S2 S1 231 2 2 0 LOGIC___2]
[ S1 S1 230 2 0 0 LOGIC___1]
[ S1 S1 232 2 0 0 LOGIC___1]
[ S2 S1 229 2 2 0 LOGIC___2]
[ S1 S1 231 2 0 0 LOGIC___1]
[ S1 S1 213 1 1 1 LOGIC___2]
[ S2 S1 228 2 2 0 LOGIC___0]
[ S1 S1 212 1 0 0 LOGIC___2]
[ S1 S1 229 2 0 0 LOGIC___1]
[ S2 S1 226 2 2 0 LOGIC___2]
[ S1 S1 228 2 0 0 LOGIC___1]
[ S2 S1 225 2 2 0 LOGIC___2]
[ S2 S1 224 2 2 0 LOGIC___2]
[ S1 S1 226 2 0 0 LOGIC___1]
[ S2 S1 223 2 2 0 LOGIC___0]
[ S1 S1 225 2 0 0 LOGIC___1]
[ S2 S1 222 2 2 0 LOGIC___0]
[ S1 S1 224 2 0 0 LOGIC___1]
[ S2 S1 221 2 2 0 LOGIC___0]
[ S1 S1 223 2 0 0 LOGIC___1]
[ S2 S1 220 2 2 0 LOGIC___2]
[ S1 S1 222 2 0 0 LOGIC___1]
[ S2 S1 219 2 2 0 LOGIC___0]
[ S2 S1 218 2 2 0 LOGIC___2]
[ S1 S1 221 2 0 0 LOGIC___1]
[ S1 S1 220 2 0 0 LOGIC___1]
[ S2 S1 217 2 2 0 LOGIC___2]
[ S1 S2 217 2 0 2 LOGIC___0]
[ S1 S1 219 2 0 0 LOGIC___2]
[ S1 S2 216 2 0 2 LOGIC___0]
[ S1 S2 215 2 0 2 LOGIC___0]
[ S1 S1 218 2 0 0 LOGIC___0]
[ S1 S1 215 2 0 0 LOGIC___0]
[ S1 S2 214 2 0 2 LOGIC___0]
[ S1 S1 217 2 0 0 LOGIC___2]
[ S1 S2 213 2 0 2 LOGIC___0]
[ S1 S1 216 2 0 0 LOGIC___2]
[ S1 S2 212 2 0 2 LOGIC___0]
[ S1 S2 211 2 0 2 LOGIC___0]
[ S1 S1 211 1 0 0 LOGIC___2]
[ S1 S1 214 2 0 0 LOGIC___2]
[ S1 S1 213 2 0 0 LOGIC___0]
[ S1 S1 212 2 0 0 LOGIC___0]
[ S1 S1 211 2 0 0 LOGIC___0]
[ S1 S2 210 2 0 2 LOGIC___0]
[ S1 S1 210 1 0 0 LOGIC___2]
[ S1 S1 210 2 0 0 LOGIC___2]
[ S1 S1 209 2 0 0 LOGIC___0]
[ S1 S2 209 2 0 2 LOGIC___0]
[ S1 S1 209 1 0 0 LOGIC___2]
[ S1 S1 208 2 0 0 LOGIC___0]
[ S1 S2 208 2 0 2 LOGIC___0]
[ S1 S2 207 2 0 2 LOGIC___0]
[ S1 S1 207 1 0 0 LOGIC___2]
[ S1 S1 207 2 0 0 LOGIC___2]
[ S1 S2 206 2 0 2 LOGIC___0]
[ S1 S1 206 1 0 0 LOGIC___2]
[ S1 S1 206 2 0 0 LOGIC___0]
[ S1 S2 205 2 0 2 LOGIC___0]
[ S1 S1 205 1 0 0 LOGIC___2]
[ S1 S1 205 2 0 0 LOGIC___2]
[ S1 S1 204 2 0 0 LOGIC___0]
[ S1 S2 204 2 0 2 LOGIC___0]
[ S1 S1 204 1 0 0 LOGIC___2]
[ S1 S1 203 2 0 0 LOGIC___0]
[ S1 S2 203 2 0 2 LOGIC___0]
[ S1 S1 203 1 0 0 LOGIC___2]
[ S1 S2 202 2 0 2 LOGIC___0]
[ S1 S1 202 1 0 0 LOGIC___2]
[ S1 S1 202 2 0 0 LOGIC___2]
[ S1 S1 201 2 0 0 LOGIC___0]
[ S1 S2 201 2 0 2 LOGIC___0]
[ S1 S1 201 1 0 0 LOGIC___2]
[ S1 S1 200 2 0 0 LOGIC___0]
[ S1 S2 200 2 0 2 LOGIC___0]
[ S1 S1 200 1 0 0 LOGIC___2]
[ S1 S1 200 1 1 1 LOGIC___2]
[ S1 S1 199 2 0 0 LOGIC___0]
[ S1 S2 199 2 0 2 LOGIC___0]
[ S1 S1 199 1 0 0 LOGIC___2]
[ S1 S1 199 1 1 1 LOGIC___2]
[ S1 S2 198 2 0 2 LOGIC___0]
[ S1 S1 198 1 0 0 LOGIC___2]
[ S1 S1 198 2 0 0 LOGIC___2]
[ S1 S2 197 2 0 2 LOGIC___0]
[ S1 S1 197 1 0 0 LOGIC___2]
[ S1 S1 197 1 1 1 LOGIC___2]
[ S1 S1 197 2 0 0 LOGIC___2]
[ S1 S2 196 2 0 2 LOGIC___0]
[ S1 S1 196 1 0 0 LOGIC___2]
[ S1 S1 196 1 1 1 LOGIC___2]
[ S1 S1 196 2 0 0 LOGIC___2]
[ S1 S2 195 2 0 2 LOGIC___0]
[ S1 S1 195 1 0 0 LOGIC___2]
[ S1 S1 195 2 0 0 LOGIC___2]
[ S1 S1 195 1 1 1 LOGIC___2]
[ S1 S2 194 2 0 2 LOGIC___0]
[ S1 S1 194 1 0 0 LOGIC___2]
[ S1 S1 194 2 0 0 LOGIC___2]
[ S1 S1 194 1 1 1 LOGIC___2]
[ S1 S2 193 2 0 2 LOGIC___0]
[ S1 S1 193 1 0 0 LOGIC___2]
[ S1 S1 193 2 0 0 LOGIC___2]
[ S1 S1 193 1 1 1 LOGIC___2]
[ S1 S2 192 2 0 2 LOGIC___0]
[ S1 S1 192 1 0 0 LOGIC___2]
[ S1 S1 192 2 0 0 LOGIC___2]
[ S1 S1 192 1 1 1 LOGIC___2]
[ S1 S2 191 2 0 2 LOGIC___0]
[ S1 S1 191 1 1 1 LOGIC___2]
[ S1 S1 191 2 0 0 LOGIC___2]
[ S1 S1 191 1 0 0 LOGIC___2]
[ S1 S2 190 2 0 2 LOGIC___0]
[ S1 S1 190 1 0 0 LOGIC___2]
[ S1 S1 190 2 0 0 LOGIC___0]
[ S1 S1 190 1 1 1 LOGIC___2]
[ S1 S2 189 2 0 2 LOGIC___0]
[ S1 S1 189 1 0 0 LOGIC___2]
[ S1 S1 189 2 0 0 LOGIC___2]
[ S1 S1 189 1 1 1 LOGIC___2]
[ S1 S2 188 2 0 2 LOGIC___0]
[ S1 S1 188 1 0 0 LOGIC___2]
[ S1 S1 188 2 0 0 LOGIC___2]
[ S1 S1 188 1 1 1 LOGIC___2]
[ S1 S1 187 2 0 0 LOGIC___0]
[ S1 S2 187 2 0 2 LOGIC___0]
[ S1 S1 187 1 0 0 LOGIC___2]
[ S1 S1 187 1 1 1 LOGIC___2]
[ S1 S1 186 2 0 0 LOGIC___2]
[ S1 S2 186 2 0 2 LOGIC___0]
[ S1 S1 186 1 0 0 LOGIC___2]
[ S1 S1 186 1 1 1 LOGIC___2]
[ S1 S2 185 2 0 2 LOGIC___0]
[ S1 S1 185 2 0 0 LOGIC___2]
[ S1 S1 185 1 0 0 LOGIC___2]
[ S1 S2 184 2 0 2 LOGIC___0]
[ S1 S1 184 1 1 1 LOGIC___2]
[ S1 S1 184 2 0 0 LOGIC___2]
[ S1 S1 184 1 0 0 LOGIC___2]
[ S1 S1 183 2 0 0 LOGIC___0]
[ S1 S2 183 2 0 2 LOGIC___0]
[ S1 S1 183 1 1 1 LOGIC___2]
[ S1 S1 183 1 0 0 LOGIC___2]
[ S1 S1 182 2 0 0 LOGIC___2]
[ S1 S2 182 2 0 2 LOGIC___0]
[ S1 S1 182 1 0 0 LOGIC___2]
[ S1 S1 182 1 1 1 LOGIC___2]
[ S1 S1 181 2 0 0 LOGIC___0]
[ S1 S2 181 2 0 2 LOGIC___0]
[ S1 S1 181 1 0 0 LOGIC___2]
[ S1 S1 181 1 1 1 LOGIC___2]
[ S1 S1 180 2 0 0 LOGIC___2]
[ S1 S2 180 2 0 2 LOGIC___0]
[ S1 S1 180 1 0 0 LOGIC___2]
[ S1 S1 180 1 1 1 LOGIC___2]
[ S1 S1 179 2 0 0 LOGIC___2]
[ S1 S2 179 2 0 2 LOGIC___0]
[ S1 S1 179 1 0 0 LOGIC___2]
[ S1 S1 179 1 1 1 LOGIC___2]
[ S1 S1 178 2 0 0 LOGIC___0]
[ S1 S2 178 2 0 2 LOGIC___0]
[ S1 S1 178 1 0 0 LOGIC___2]
[ S1 S1 178 1 1 1 LOGIC___2]
[ S1 S1 177 2 0 0 LOGIC___2]
[ S1 S2 177 2 0 2 LOGIC___0]
[ S1 S1 177 1 0 0 LOGIC___2]
[ S1 S1 177 1 1 1 LOGIC___2]
[ S1 S1 176 2 0 0 LOGIC___2]
[ S1 S2 176 2 0 2 LOGIC___0]
[ S1 S1 176 1 0 0 LOGIC___2]
[ S1 S1 176 1 1 1 LOGIC___2]
[ S1 S1 175 2 0 0 LOGIC___2]
[ S1 S2 175 2 0 2 LOGIC___0]
[ S1 S1 175 1 1 1 LOGIC___2]
[ S1 S1 175 1 0 0 LOGIC___2]
[ S1 S1 174 2 0 0 LOGIC___2]
[ S1 S2 174 2 0 2 LOGIC___0]
[ S1 S1 174 1 1 1 LOGIC___2]
[ S1 S1 174 1 0 0 LOGIC___2]
[ S1 S1 173 2 0 0 LOGIC___0]
[ S1 S2 173 2 0 2 LOGIC___0]
[ S1 S1 173 1 0 0 LOGIC___2]
[ S1 S1 173 1 1 1 LOGIC___2]
[ S1 S1 172 2 0 0 LOGIC___2]
[ S1 S2 172 2 0 2 LOGIC___0]
[ S1 S1 172 1 0 0 LOGIC___2]
[ S1 S1 171 2 0 0 LOGIC___2]
[ S1 S2 171 2 0 2 LOGIC___0]
[ S1 S1 171 1 1 1 LOGIC___2]
[ S1 S1 171 1 0 0 LOGIC___2]
[ S1 S1 170 2 0 0 LOGIC___2]
```

```
[ S1 S2 170 2 0 2 LOGIC__0]
[ S1 S1 170 1 1 1 LOGIC__2]
[ S1 S1 170 1 0 0 LOGIC__2]
[ S1 S1 169 2 0 0 LOGIC__2]
[ S1 S2 169 2 0 2 LOGIC__0]
[ S1 S1 169 1 1 1 LOGIC__2]
[ S1 S1 169 1 0 0 LOGIC__2]
[ S1 S1 168 2 0 0 LOGIC__2]
[ S1 S2 168 2 0 2 LOGIC__0]
[ S1 S1 168 1 0 0 LOGIC__2]
[ S1 S1 167 2 0 0 LOGIC__2]
[ S1 S2 167 2 0 2 LOGIC__0]
[ S1 S1 167 1 0 0 LOGIC__2]
[ S1 S1 167 1 1 1 LOGIC__2]
[ S1 S1 166 2 0 0 LOGIC__2]
[ S1 S2 166 2 0 2 LOGIC__0]
[ S1 S1 166 1 0 0 LOGIC__2]
[ S1 S1 166 1 1 1 LOGIC__2]
[ S1 S1 165 2 0 0 LOGIC__2]
[ S1 S2 165 2 0 2 LOGIC__0]
[ S1 S1 165 1 1 1 LOGIC__2]
[ S1 S1 165 1 0 0 LOGIC__2]
[ S1 S1 164 2 0 0 LOGIC__2]
[ S1 S2 164 2 0 2 LOGIC__0]
[ S1 S1 164 1 0 0 LOGIC__2]
[ S1 S1 164 1 1 1 LOGIC__2]
[ S1 S1 163 2 0 0 LOGIC__0]
[ S1 S2 163 2 0 2 LOGIC__0]
[ S1 S1 163 1 1 1 LOGIC__2]
[ S1 S1 163 1 0 0 LOGIC__2]
[ S1 S1 162 2 0 0 LOGIC__2]
[ S1 S2 162 2 0 2 LOGIC__0]
[ S1 S1 162 1 1 1 LOGIC__2]
[ S1 S1 162 1 0 0 LOGIC__2]
[ S1 S1 161 2 0 0 LOGIC__2]
[ S1 S2 161 2 0 2 LOGIC__0]
[ S1 S1 161 1 1 1 LOGIC__2]
[ S1 S1 161 1 0 0 LOGIC__2]
[ S1 S1 160 2 0 0 LOGIC__2]
[ S1 S2 160 2 0 2 LOGIC__0]
[ S1 S1 160 1 0 0 LOGIC__2]
[ S1 S1 160 1 1 1 LOGIC__2]
[ S1 S1 159 2 0 0 LOGIC__2]
[ S1 S2 159 2 0 2 LOGIC__0]
[ S1 S1 159 1 0 0 LOGIC__2]
[ S1 S1 158 2 0 0 LOGIC__2]
[ S1 S2 158 2 0 2 LOGIC__0]
[ S1 S1 158 1 0 0 LOGIC__2]
[ S1 S1 157 2 0 0 LOGIC__2]
[ S1 S2 157 2 0 2 LOGIC__0]
[ S1 S1 157 1 0 0 LOGIC__2]
[ S1 S1 156 2 0 0 LOGIC__0]
[ S1 S2 156 2 0 2 LOGIC__0]
[ S1 S1 156 1 0 0 LOGIC__2]
[ S1 S1 155 2 0 0 LOGIC__0]
[ S1 S2 155 2 0 2 LOGIC__0]
[ S1 S1 155 1 0 0 LOGIC__2]
[ S1 S1 154 2 0 0 LOGIC__2]
[ S1 S2 154 2 0 2 LOGIC__0]
[ S1 S1 154 1 0 0 LOGIC__2]
[ S1 S1 153 2 0 0 LOGIC__2]
[ S1 S2 153 2 0 2 LOGIC__0]
[ S1 S1 153 1 0 0 LOGIC__2]
[ S1 S1 152 2 0 0 LOGIC__0]
[ S1 S2 152 2 0 2 LOGIC__0]
[ S1 S1 152 1 0 0 LOGIC__2]
[ S1 S1 151 2 0 0 LOGIC__2]
[ S1 S2 151 2 0 2 LOGIC__0]
[ S1 S1 151 1 0 0 LOGIC__2]
] -> Example_G5N;
id3(Example_G5N,Attribute_G5N,
Class_G5N)->Rule_G5N;
print_rule(Rule_G5N);
pf(Rule_G5N,6);nl(1);
vars Pruning Level;
true -> Pruning;
99 -> Level;
id3_chisqd(Example_G5N,Attribut
e_G5N,Class_G5N,
        Level,Pruning) ->Rule_G5N;
print_rule(Rule_G5N);
pfx(Rule_G5N,6);nl(1);


;;;____________________76
mm Guns Learning Samples______________
_
[[LOGICAL SURG76_L_1__1_R_
S0 S1 S2 S5]
 [INTEGER ANGLE]
  [INTEGER PRIOR_L_NUMT-
GT_R_]
   [INTEGER PRIOR_L_TGT-
G76_L_1_R__R_]
 ] -> Attribute_G76;
[CLASSES G76_ALLOCATE
  LOGIC__1 LOGIC__0] ->
Class_G76;
[
[ S1 150 1 0 LOGIC__0]
[ S1 149 1 0 LOGIC__0]
[ S1 148 1 0 LOGIC__0]
[ S1 147 1 0 LOGIC__0]
[ S1 146 1 0 LOGIC__0]
[ S1 140 1 0 LOGIC__0]
[ S1 134 1 0 LOGIC__0]
[ S1 132 1 0 LOGIC__0]
[ S1 136 1 0 LOGIC__0]
[ S1 128 1 0 LOGIC__0]
[ S1 126 1 0 LOGIC__0]
[ S1 131 1 0 LOGIC__0]
[ S1 129 1 0 LOGIC__0]
[ S1 124 1 0 LOGIC__0]
[ S1 127 1 0 LOGIC__0]
[ S1 123 1 0 LOGIC__0]
[ S1 125 1 0 LOGIC__0]
[ S1 119 1 0 LOGIC__0]
[ S1 121 1 0 LOGIC__0]
[ S1 122 1 0 LOGIC__0]
[ S1 118 1 0 LOGIC__0]
[ S1 117 1 0 LOGIC__0]
[ S1 120 1 0 LOGIC__0]
[ S1 116 1 0 LOGIC__0]
[ S1 115 1 0 LOGIC__0]
[ S1 114 1 0 LOGIC__0]
[ S1 111 1 0 LOGIC__0]
[ S1 112 1 0 LOGIC__0]
[ S1 110 1 0 LOGIC__0]
[ S1 113 1 0 LOGIC__0]
[ S1 109 1 0 LOGIC__0]
[ S1 108 1 0 LOGIC__0]
[ S1 107 1 0 LOGIC__0]
[ S1 106 1 0 LOGIC__0]
[ S1 105 1 0 LOGIC__0]
[ S1 102 1 0 LOGIC__1]
[ S2 104 1 1 LOGIC__0]
[ S1 101 1 0 LOGIC__1]
[ S1 100 1 0 LOGIC__1]
[ S2 102 1 1 LOGIC__0]
[ S1 99 1 0 LOGIC__1]
[ S1 98 1 0 LOGIC__0]
[ S1 97 1 0 LOGIC__1]
[ S2 99 1 1 LOGIC__0]
[ S2 98 1 1 LOGIC__0]
[ S1 96 1 0 LOGIC__1]
[ S1 95 1 0 LOGIC__1]
[ S2 97 1 1 LOGIC__0]
[ S1 94 1 0 LOGIC__1]
[ S2 96 1 1 LOGIC__0]
[ S2 95 1 1 LOGIC__0]
[ S1 93 1 0 LOGIC__1]
[ S2 94 1 1 LOGIC__0]
[ S1 92 1 0 LOGIC__1]
[ S1 91 1 0 LOGIC__1]
[ S2 93 1 1 LOGIC__0]
[ S2 92 1 1 LOGIC__0]
[ S1 90 1 0 LOGIC__1]
[ S2 91 1 1 LOGIC__0]
[ S2 89 1 1 LOGIC__0]
[ S2 88 1 1 LOGIC__0]
[ S1 89 1 0 LOGIC__1]
[ S1 88 1 0 LOGIC__1]
[ S2 87 1 1 LOGIC__0]
[ S2 86 1 1 LOGIC__0]
[ S1 87 1 0 LOGIC__1]
[ S1 144 2 0 LOGIC__0]
[ S1 86 1 0 LOGIC__1]
[ S2 85 1 1 LOGIC__0]
[ S1 85 1 0 LOGIC__1]
[ S2 84 1 1 LOGIC__0]
[ S2 83 1 1 LOGIC__0]
[ S1 84 1 0 LOGIC__1]
[ S1 83 1 0 LOGIC__1]
[ S2 82 1 1 LOGIC__0]
[ S1 82 1 0 LOGIC__1]
[ S2 81 1 1 LOGIC__0]
[ S1 81 1 0 LOGIC__1]
[ S2 80 1 1 LOGIC__0]
[ S1 80 1 0 LOGIC__1]
[ S2 79 1 1 LOGIC__0]
[ S2 78 1 1 LOGIC__0]
[ S1 79 1 0 LOGIC__1]
[ S1 78 1 0 LOGIC__1]
[ S2 77 1 1 LOGIC__0]
[ S1 77 1 0 LOGIC__1]
[ S2 76 1 1 LOGIC__0]
[ S1 76 1 0 LOGIC__0]
[ S1 75 1 0 LOGIC__0]
[ S2 74 1 1 LOGIC__0]
[ S1 74 1 0 LOGIC__1]
[ S2 73 1 1 LOGIC__0]
```

```
[S1 73 1 0 LOGIC___1]
[S2 72 1 1 LOGIC___0]
[S1 72 1 0 LOGIC___0]
[S1 71 1 0 LOGIC___0]
[S2 71 1 1 LOGIC___0]
[S2 70 1 1 LOGIC___0]
[S1 70 1 0 LOGIC___1]
[S2 69 1 1 LOGIC___0]
[S1 69 1 0 LOGIC___1]
[S2 68 1 1 LOGIC___0]
[S1 68 1 0 LOGIC___1]
[S2 67 1 1 LOGIC___0]
[S1 67 1 0 LOGIC___1]
[S2 66 1 1 LOGIC___0]
[S1 66 1 0 LOGIC___1]
[S2 65 1 1 LOGIC___0]
[S1 65 1 0 LOGIC___1]
[S2 64 1 1 LOGIC___0]
[S1 64 1 0 LOGIC___1]
[S2 63 1 1 LOGIC___0]
[S1 63 1 0 LOGIC___1]
[S2 62 1 1 LOGIC___0]
[S1 62 1 0 LOGIC___1]
[S2 61 1 1 LOGIC___0]
[S1 61 1 0 LOGIC___1]
[S2 60 1 1 LOGIC___0]
[S1 60 1 0 LOGIC___1]
[S2 59 1 1 LOGIC___0]
[S1 59 1 0 LOGIC___1]
[S2 58 1 1 LOGIC___0]
[S1 58 1 0 LOGIC___1]
[S2 57 1 1 LOGIC___0]
[S1 57 1 0 LOGIC___1]
[S2 56 1 1 LOGIC___0]
[S1 56 1 0 LOGIC___1]
[S2 55 1 1 LOGIC___0]
[S1 55 1 0 LOGIC___1]
[S2 54 1 1 LOGIC___0]
[S1 54 1 0 LOGIC___1]
[S2 53 1 1 LOGIC___0]
[S1 53 1 0 LOGIC___1]
[S1 52 1 0 LOGIC___1]
[S2 52 1 1 LOGIC___0]
[S2 51 1 1 LOGIC___0]
[S1 51 1 0 LOGIC___1]
[S2 50 1 1 LOGIC___0]
[S1 50 1 0 LOGIC___1]
[S2 49 1 1 LOGIC___0]
[S1 49 1 0 LOGIC___1]
[S2 48 1 1 LOGIC___0]
[S1 48 1 0 LOGIC___1]
[S2 47 1 1 LOGIC___0]
[S1 47 1 0 LOGIC___1]
[S2 46 1 1 LOGIC___0]
[S1 46 1 0 LOGIC___1]
[S2 45 1 1 LOGIC___0]
[S1 45 1 0 LOGIC___1]
[S2 44 1 1 LOGIC___0]
[S1 44 1 0 LOGIC___1]
[S2 43 1 1 LOGIC___0]
[S1 43 1 0 LOGIC___1]
[S2 42 1 1 LOGIC___0]
[S1 42 1 0 LOGIC___0]
[S1 41 1 0 LOGIC___1]
[S2 40 1 1 LOGIC___0]
[S1 40 1 0 LOGIC___1]
[S2 39 1 1 LOGIC___0]
[S1 39 1 0 LOGIC___1]
[S2 38 1 1 LOGIC___0]
[S1 38 1 0 LOGIC___1]
[S2 37 1 1 LOGIC___0]
[S1 37 1 0 LOGIC___0]
[S1 36 1 0 LOGIC___1]
[S2 36 1 1 LOGIC___0]
[S2 35 1 1 LOGIC___0]
[S1 35 1 0 LOGIC___1]
[S1 34 1 0 LOGIC___1]
[S2 34 1 1 LOGIC___0]
[S2 33 1 1 LOGIC___0]
[S1 33 1 0 LOGIC___1]
[S1 32 1 0 LOGIC___1]
[S2 32 1 1 LOGIC___0]
[S2 31 1 1 LOGIC___0]
[S1 31 1 0 LOGIC___1]
[S1 30 1 0 LOGIC___1]
[S2 30 1 1 LOGIC___0]
[S2 29 1 1 LOGIC___0]
[S1 29 1 0 LOGIC___1]
[S1 28 1 0 LOGIC___0]
[S2 27 1 1 LOGIC___0]
[S1 27 1 0 LOGIC___1]
[S1 26 1 0 LOGIC___1]
[S2 26 1 1 LOGIC___0]
[S2 25 1 1 LOGIC___0]
[S1 25 1 0 LOGIC___1]
[S1 24 1 0 LOGIC___1]
[S2 24 1 1 LOGIC___0]
[S2 23 1 1 LOGIC___0]
[S1 23 1 0 LOGIC___1]
[S1 22 1 0 LOGIC___1]
[S2 22 1 1 LOGIC___0]
[S2 21 1 1 LOGIC___0]
[S1 21 1 0 LOGIC___1]
[S1 20 1 0 LOGIC___1]
[S2 20 1 1 LOGIC___0]
[S2 19 1 1 LOGIC___0]
[S1 19 1 0 LOGIC___1]
[S1 18 1 0 LOGIC___1]
[S2 18 1 1 LOGIC___0]
[S1 17 1 0 LOGIC___1]
[S2 17 1 1 LOGIC___0]
[S1 16 1 0 LOGIC___1]
[S2 16 1 1 LOGIC___0]
[S1 15 1 0 LOGIC___1]
[S2 15 1 1 LOGIC___0]
[S1 14 1 0 LOGIC___1]
[S2 14 1 1 LOGIC___0]
[S1 13 1 0 LOGIC___1]
[S2 13 1 1 LOGIC___0]
[S1 12 1 0 LOGIC___1]
[S2 12 1 1 LOGIC___0]
[S1 11 1 0 LOGIC___1]
[S2 11 1 1 LOGIC___0]
[S1 10 1 0 LOGIC___1]
[S2 10 1 1 LOGIC___0]
[S1 32 1 1 LOGIC___1]
[S1 9 1 0 LOGIC___1]
[S2 9 1 1 LOGIC___0]
[S2 8 1 1 LOGIC___0]
[S1 8 1 0 LOGIC___1]
[S1 7 1 0 LOGIC___1]
[S2 7 1 1 LOGIC___0]
[S1 17 1 1 LOGIC___1]
[S2 6 1 1 LOGIC___0]
[S1 13 1 1 LOGIC___1]
[S1 6 1 0 LOGIC___1]
[S1 5 1 0 LOGIC___1]
[S2 5 1 1 LOGIC___0]
[S1 22 1 1 LOGIC___1]
[S1 4 1 0 LOGIC___1]
[S2 4 1 1 LOGIC___0]
[S1 19 1 1 LOGIC___1]
[S1 3 1 0 LOGIC___1]
[S2 3 1 1 LOGIC___0]
[S1 2 1 0 LOGIC___1]
[S2 2 1 1 LOGIC___0]
[S1 1 1 0 LOGIC___1]
[S2 1 1 1 LOGIC___0]
[S1 2 1 1 LOGIC___1]
[S1 1 1 1 LOGIC___1]
[S1 360 1 0 LOGIC___1]
[S2 360 1 1 LOGIC___0]
[S1 359 1 0 LOGIC___1]
[S2 359 1 1 LOGIC___0]
[S1 358 1 0 LOGIC___1]
[S2 358 1 1 LOGIC___0]
[S1 357 1 0 LOGIC___1]
[S2 357 1 1 LOGIC___0]
[S1 209 1 1 LOGIC___0]
[S1 356 1 0 LOGIC___1]
[S2 356 1 1 LOGIC___0]
[S1 355 1 0 LOGIC___1]
[S2 355 1 1 LOGIC___0]
[S1 351 1 1 LOGIC___1]
[S1 353 1 0 LOGIC___1]
[S1 354 1 0 LOGIC___1]
[S2 354 1 1 LOGIC___0]
[S1 347 1 1 LOGIC___1]
[S2 353 1 1 LOGIC___0]
[S1 352 1 0 LOGIC___1]
[S2 352 1 1 LOGIC___0]
[S1 351 1 0 LOGIC___1]
[S1 348 1 0 LOGIC___1]
[S1 350 1 0 LOGIC___1]
[S2 350 1 1 LOGIC___0]
[S1 347 1 0 LOGIC___1]
[S1 349 1 0 LOGIC___1]
[S2 349 1 1 LOGIC___0]
[S1 346 1 0 LOGIC___1]
[S2 348 1 1 LOGIC___0]
[S1 344 1 0 LOGIC___1]
[S2 347 1 1 LOGIC___0]
[S2 346 1 1 LOGIC___0]
[S1 341 1 0 LOGIC___1]
[S1 345 1 0 LOGIC___1]
[S2 345 1 1 LOGIC___0]
[S2 344 1 1 LOGIC___0]
[S1 325 1 0 LOGIC___1]
[S1 340 1 0 LOGIC___1]
[S1 343 1 0 LOGIC___1]
[S2 343 1 1 LOGIC___0]
[S1 296 1 1 LOGIC___1]
```

[S1 342 1 0 LOGIC___1]
[S2 342 1 1 LOGIC___0]
[S1 332 1 0 LOGIC___1]
[S1 241 1 0 LOGIC___0]
[S2 341 1 1 LOGIC___0]
[S1 329 1 0 LOGIC___1]
[S2 340 1 1 LOGIC___0]
[S1 289 1 0 LOGIC___1]
[S1 339 1 0 LOGIC___1]
[S2 339 1 1 LOGIC___0]
[S1 330 1 0 LOGIC___1]
[S1 338 1 0 LOGIC___1]
[S1 334 1 0 LOGIC___1]
[S2 338 1 1 LOGIC___0]
[S1 336 1 0 LOGIC___1]
[S1 337 1 0 LOGIC___1]
[S2 337 1 1 LOGIC___0]
[S1 284 1 0 LOGIC___1]
[S1 335 1 0 LOGIC___1]
[S2 336 1 1 LOGIC___0]
[S1 321 1 0 LOGIC___1]
[S2 335 1 1 LOGIC___0]
[S1 333 1 0 LOGIC___1]
[S2 334 1 1 LOGIC___0]
[S2 333 1 1 LOGIC___0]
[S1 322 1 0 LOGIC___1]
[S1 324 1 0 LOGIC___1]
[S2 332 1 1 LOGIC___0]
[S1 323 1 0 LOGIC___1]
[S1 331 1 0 LOGIC___1]
[S2 331 1 1 LOGIC___0]
[S2 329 1 1 LOGIC___0]
[S1 320 1 0 LOGIC___1]
[S1 328 1 0 LOGIC___1]
[S1 316 1 0 LOGIC___1]
[S1 327 1 0 LOGIC___1]
[S2 327 1 1 LOGIC___0]
[S1 326 1 0 LOGIC___1]
[S1 317 1 0 LOGIC___1]
[S2 326 1 1 LOGIC___0]
[S2 325 1 1 LOGIC___0]
[S1 319 1 0 LOGIC___1]
[S1 279 1 0 LOGIC___1]
[S2 324 1 1 LOGIC___0]
[S1 313 1 0 LOGIC___1]
[S2 323 1 1 LOGIC___0]
[S1 303 1 0 LOGIC___1]
[S1 315 1 0 LOGIC___1]
[S2 322 1 1 LOGIC___0]
[S1 275 1 0 LOGIC___1]
[S1 300 1 0 LOGIC___1]
[S2 320 1 1 LOGIC___0]
[S1 312 1 0 LOGIC___1]
[S1 318 1 0 LOGIC___1]
[S2 319 1 1 LOGIC___0]
[S2 318 1 1 LOGIC___0]
[S1 309 1 0 LOGIC___1]
[S2 317 1 1 LOGIC___0]
[S1 304 1 0 LOGIC___1]
[S2 316 1 1 LOGIC___0]
[S1 314 1 0 LOGIC___1]
[S1 307 1 0 LOGIC___1]
[S2 315 1 1 LOGIC___0]
[S1 302 1 0 LOGIC___1]
[S2 314 1 1 LOGIC___0]
[S1 301 1 0 LOGIC___1]
[S2 313 1 1 LOGIC___0]
[S2 312 1 1 LOGIC___0]
[S1 298 1 0 LOGIC___1]
[S1 311 1 0 LOGIC___1]
[S1 310 1 0 LOGIC___1]
[S2 311 1 1 LOGIC___0]
[S1 308 1 0 LOGIC___1]
[S2 310 1 1 LOGIC___0]
[S2 309 1 1 LOGIC___0]
[S1 294 1 0 LOGIC___1]
[S2 308 1 1 LOGIC___0]
[S1 299 1 0 LOGIC___1]
[S1 281 1 0 LOGIC___1]
[S1 306 1 0 LOGIC___1]
[S2 307 1 1 LOGIC___0]
[S1 280 1 0 LOGIC___1]
[S2 306 1 1 LOGIC___0]
[S1 287 1 0 LOGIC___1]
[S1 305 1 0 LOGIC___1]
[S1 296 1 0 LOGIC___1]
[S2 305 1 1 LOGIC___0]
[S1 278 1 0 LOGIC___1]
[S2 304 1 1 LOGIC___0]
[S1 285 1 0 LOGIC___1]
[S1 271 1 0 LOGIC___1]
[S2 302 1 1 LOGIC___0]
[S1 252 1 0 LOGIC___0]
[S1 273 1 0 LOGIC___1]
[S2 301 1 1 LOGIC___0]
[S2 300 1 1 LOGIC___0]
[S1 283 1 0 LOGIC___1]
[S2 299 1 1 LOGIC___0]
[S1 282 1 0 LOGIC___1]
[S1 297 1 0 LOGIC___1]
[S2 298 1 1 LOGIC___0]
[S1 231 1 0 LOGIC___0]
[S2 297 1 1 LOGIC___0]
[S1 295 1 0 LOGIC___1]
[S2 296 1 1 LOGIC___0]
[S1 293 1 0 LOGIC___1]
[S2 295 1 1 LOGIC___0]
[S2 294 1 1 LOGIC___0]
[S1 208 1 0 LOGIC___0]
[S1 292 1 0 LOGIC___1]
[S2 293 1 1 LOGIC___0]
[S1 260 1 0 LOGIC___1]
[S1 291 1 0 LOGIC___1]
[S2 292 1 1 LOGIC___0]
[S1 290 1 0 LOGIC___1]
[S2 291 1 1 LOGIC___0]
[S1 272 1 0 LOGIC___1]
[S2 290 1 1 LOGIC___0]
[S1 288 1 0 LOGIC___1]
[S2 289 1 1 LOGIC___0]
[S1 240 1 0 LOGIC___0]
[S1 286 1 0 LOGIC___1]
[S2 288 1 1 LOGIC___0]
[S1 227 2 0 LOGIC___0]
[S2 286 1 1 LOGIC___0]
[S1 267 1 0 LOGIC___1]
[S1 254 1 0 LOGIC___0]
[S2 285 1 1 LOGIC___0]
[S1 274 1 0 LOGIC___1]
[S2 284 1 1 LOGIC___0]
[S1 265 1 0 LOGIC___1]
[S1 263 1 0 LOGIC___1]
[S2 283 1 1 LOGIC___0]
[S1 262 1 0 LOGIC___1]
[S2 282 1 1 LOGIC___0]
[S2 281 1 1 LOGIC___0]
[S2 280 1 1 LOGIC___0]
[S1 248 1 0 LOGIC___0]
[S2 279 1 1 LOGIC___0]
[S1 246 1 0 LOGIC___0]
[S1 277 1 0 LOGIC___1]
[S2 278 1 1 LOGIC___0]
[S1 268 1 0 LOGIC___1]
[S1 276 1 0 LOGIC___1]
[S2 277 1 1 LOGIC___0]
[S1 256 1 0 LOGIC___1]
[S2 276 1 1 LOGIC___0]
[S1 255 1 0 LOGIC___0]
[S2 275 1 1 LOGIC___0]
[S2 274 1 1 LOGIC___0]
[S2 273 1 1 LOGIC___0]
[S2 272 1 1 LOGIC___0]
[S1 270 1 0 LOGIC___1]
[S2 271 1 1 LOGIC___0]
[S1 261 1 0 LOGIC___1]
[S1 269 1 0 LOGIC___1]
[S2 269 1 1 LOGIC___0]
[S2 268 1 1 LOGIC___0]
[S1 266 1 0 LOGIC___1]
[S2 267 1 1 LOGIC___0]
[S1 257 1 0 LOGIC___1]
[S2 266 1 1 LOGIC___0]
[S2 265 1 1 LOGIC___0]
[S1 264 1 0 LOGIC___1]
[S2 264 1 1 LOGIC___0]
[S2 263 1 1 LOGIC___0]
[S2 262 1 1 LOGIC___0]
[S2 261 1 1 LOGIC___0]
[S2 259 1 1 LOGIC___0]
[S1 237 1 0 LOGIC___0]
[S1 258 1 0 LOGIC___1]
[S1 259 1 0 LOGIC___1]
[S2 255 1 1 LOGIC___0]
[S2 254 1 1 LOGIC___0]
[S1 253 1 0 LOGIC___0]
[S1 251 1 0 LOGIC___0]
[S1 250 1 0 LOGIC___0]
[S1 249 1 0 LOGIC___0]
[S1 247 1 0 LOGIC___0]
[S1 245 1 0 LOGIC___0]
[S1 244 1 0 LOGIC___0]
[S1 243 1 0 LOGIC___0]
[S1 242 1 0 LOGIC___0]
[S1 238 1 0 LOGIC___0]
[S1 235 1 0 LOGIC___0]
[S1 233 1 0 LOGIC___0]
[S1 236 1 0 LOGIC___0]
[S1 234 1 0 LOGIC___0]
[S1 226 1 0 LOGIC___0]
[S1 224 1 0 LOGIC___0]
[S1 229 1 0 LOGIC___0]
[S1 225 1 0 LOGIC___0]

```
[ S1 220 1 0 LOGIC___0]
[ S1 213 1 0 LOGIC___0]
[ S1 214 1 0 LOGIC___0]
[ S1 212 1 0 LOGIC___0]
[ S1 211 1 0 LOGIC___0]
[ S1 210 1 0 LOGIC___0]
[ S1 209 1 0 LOGIC___0]
[ S1 207 1 0 LOGIC___0]
[ S1 206 1 0 LOGIC___0]
[ S1 205 1 0 LOGIC___0]
[ S1 204 1 0 LOGIC___0]
[ S1 203 1 0 LOGIC___0]
[ S1 202 1 0 LOGIC___0]
[ S1 201 1 0 LOGIC___0]
[ S1 200 1 0 LOGIC___0]
[ S1 199 1 0 LOGIC___0]
[ S1 198 1 0 LOGIC___0]
[ S1 197 1 0 LOGIC___0]
[ S1 196 1 0 LOGIC___0]
[ S1 195 1 0 LOGIC___0]
[ S1 194 1 0 LOGIC___0]
[ S1 193 1 0 LOGIC___0]
[ S1 192 1 0 LOGIC___0]
[ S1 191 1 0 LOGIC___0]
[ S1 190 1 0 LOGIC___0]
[ S1 189 1 0 LOGIC___0]
[ S1 188 1 0 LOGIC___0]
[ S1 187 1 0 LOGIC___0]
[ S1 186 1 0 LOGIC___0]
[ S1 185 1 0 LOGIC___0]
[ S1 184 1 0 LOGIC___0]
[ S1 183 1 0 LOGIC___0]
[ S1 182 1 0 LOGIC___0]
[ S1 181 1 0 LOGIC___0]
[ S1 180 1 0 LOGIC___0]
[ S1 179 1 0 LOGIC___0]
[ S1 178 1 0 LOGIC___0]
[ S1 177 1 0 LOGIC___0]
[ S1 176 1 0 LOGIC___0]
[ S1 175 1 0 LOGIC___0]
[ S1 174 1 0 LOGIC___0]
[ S1 173 1 0 LOGIC___0]
[ S1 172 1 0 LOGIC___0]
[ S1 171 1 0 LOGIC___0]
[ S1 170 1 0 LOGIC___0]
[ S1 169 1 0 LOGIC___0]
[ S1 168 1 0 LOGIC___0]
[ S1 167 1 0 LOGIC___0]
[ S1 166 1 0 LOGIC___0]
[ S1 165 1 0 LOGIC___0]
[ S1 164 1 0 LOGIC___0]
[ S1 163 1 0 LOGIC___0]
[ S1 162 1 0 LOGIC___0]
[ S1 161 1 0 LOGIC___0]
[ S1 160 1 0 LOGIC___0]
[ S1 159 1 0 LOGIC___0]
[ S1 158 1 0 LOGIC___0]
[ S1 157 1 0 LOGIC___0]
[ S1 156 1 0 LOGIC___0]
[ S1 155 1 0 LOGIC___0]
[ S1 154 1 0 LOGIC___0]
[ S1 153 1 0 LOGIC___0]
[ S1 152 1 0 LOGIC___0]
[ S1 151 1 0 LOGIC___0]
] -> Example_G76;
id3(Example_G76,Attribute_G76,C
lass_G76)->Rule_G76;
print_rule(Rule_G76);
pf(Rule_G76,6);nl(1);
vars Pruning Level;
true -> Pruning;
99 -> Level;
id3_chisqd(Example_G76,Attribute
_G76,Class_G76,
        Level,Pruning) ->Rule_G76;
print_rule(Rule_G76);
pfx(Rule_G76,6);nl(1);

;;;________________________40
mm  Guns  Learning  Sam-
ples_______________________
_
[[LOGICAL SURG40_L_1__1_R_
S1 S2 S5]
 [LOGICAL SURG40_L_1__2_R_
S1 S2 S5]
 [INTEGER ANGLE]
  [INTEGER  PRIOR_L_NUMT-
GT_R_]
   [INTEGER    PRIOR_L_TGT-
G40_L_1_R__R_]
   [INTEGER    PRIOR_L_TGT-
G40_L_2_R__R_]
 ] -> Attribute_G40;
[CLASSES G40_ALLOCATE
     LOGIC___1    LOGIC___2
LOGIC___0] -> Class_G40;
[
[ S1 S1 150 1 1 1 LOGIC___1]
[ S2 S1 150 1 1 1 LOGIC___0]
[ S1 S1 149 1 1 1 LOGIC___1]
[ S2 S1 149 1 1 1 LOGIC___0]
[ S1 S1 148 1 1 1 LOGIC___1]
[ S1 S1 147 1 1 ! LOGIC___1]
[ S2 S1 148 1 1 1 LOGIC___0]
[ S1 S1 147 1 2 2 LOGIC___0]
[ S1 S1 148 1 2 2 LOGIC___0]
[ S1 S1 146 1 1 1 LOGIC___1]
[ S1 S1 145 1 1 1 LOGIC___1]
[ S2 S1 146 1 1 1 LOGIC___0]
[ S2 S1 147 1 1 1 LOGIC___0]
[ S2 S1 145 1 1 1 LOGIC___0]
[ S1 S1 144 1 1 1 LOGIC___1]
[ S1 S1 151 1 1 1 LOGIC___1]
[ S2 S1 151 1 1 1 LOGIC___0]
[ S1 S1 152 1 1 1 LOGIC___1]
[ S2 S1 152 1 1 1 LOGIC___0]
[ S2 S1 153 1 1 1 LOGIC___0]
[ S1 S1 153 1 1 1 LOGIC___1]
[ S1 S1 154 1 1 1 LOGIC___1]
[ S1 S1 110 1 1 1 LOGIC___1]
[ S1 S1 99 1 1 1 LOGIC___1]
[ S5 S1 153 1 1 1 LOGIC___0]
[ S1 S1 100 1 1 1 LOGIC___1]
[ S2 S1 154 1 1 1 LOGIC___0]
[ S1 S1 95 1 1 1 LOGIC___1]
[ S1 S1 155 1 1 1 LOGIC___1]
[ S2 S1 155 1 1 1 LOGIC___0]
[ S1 S1 93 1 1 1 LOGIC___1]
[ S1 S1 88 1 1 1 LOGIC___1]
[ S1 S1 86 1 1 1 LOGIC___1]
[ S1 S1 156 1 1 1 LOGIC___1]
[ S2 S1 156 1 1 1 LOGIC___0]
[ S1 S1 83 1 1 1 LOGIC___1]
[ S1 S1 80 1 1 1 LOGIC___1]
[ S1 S1 157 1 1 1 LOGIC___1]
[ S2 S1 78 1 1 1 LOGIC___0]
[ S1 S1 107 1 1 1 LOGIC___1]
[ S1 S1 78 1 1 1 LOGIC___1]
[ S2 S1 76 1 1 1 LOGIC___0]
[ S5 S1 157 1 1 1 LOGIC___0]
[ S1 S1 75 1 1 1 LOGIC___1]
[ S2 S1 73 1 1 1 LOGIC___0]
[ S1 S1 72 1 1 1 LOGIC___1]
[ S1 S1 158 1 1 1 LOGIC___1]
[ S1 S1 71 1 1 1 LOGIC___1]
[ S1 S1 159 1 1 1 LOGIC___1]
[ S1 S1 68 1 1 1 LOGIC___1]
[ S2 S1 159 1 1 1 LOGIC___0]
[ S1 S1 66 1 1 1 LOGIC___1]
[ S1 S1 65 1 1 1 LOGIC___1]
[ S1 S1 64 1 1 1 LOGIC___1]
[ S2 S1 63 1 1 1 LOGIC___0]
[ S2 S1 160 1 1 1 LOGIC___0]
[ S1 S1 160 1 1 1 LOGIC___1]
[ S1 S1 161 1 1 1 LOGIC___0]
[ S1 S1 59 1 1 1 LOGIC___1]
[ S1 S1 58 1 1 1 LOGIC___1]
[ S2 S1 161 1 1 1 LOGIC___0]
[ S1 S1 56 1 1 1 LOGIC___1]
[ S1 S1 55 1 1 1 LOGIC___1]
[ S1 S1 161 1 2 2 LOGIC___0]
[ S1 S1 162 1 2 2 LOGIC___0]
[ S1 S1 53 1 1 1 LOGIC___1]
[ S1 S1 52 1 1 1 LOGIC___1]
[ S1 S1 50 1 1 1 LOGIC___1]
[ S1 S1 162 1 1 1 LOGIC___0]
[ S1 S1 163 1 2 2 LOGIC___0]
[ S1 S1 49 1 1 1 LOGIC___1]
[ S1 S1 163 1 1 1 LOGIC___0]
[ S1 S1 48 1 1 1 LOGIC___1]
[ S1 S1 47 1 1 1 LOGIC___1]
[ S2 S1 46 1 1 1 LOGIC___0]
[ S1 S1 60 1 1 1 LOGIC___1]
[ S1 S1 164 1 1 1 LOGIC___0]
[ S1 S1 46 1 1 1 LOGIC___1]
[ S2 S1 45 1 1 1 LOGIC___0]
[ S1 S1 51 1 1 1 LOGIC___1]
[ S1 S1 44 1 1 1 LOGIC___1]
[ S1 S1 164 1 2 2 LOGIC___0]
[ S1 S1 43 1 1 1 LOGIC___1]
[ S1 S1 165 1 2 2 LOGIC___0]
[ S1 S1 42 1 1 1 LOGIC___1]
[ S2 S1 41 1 1 1 LOGIC___0]
[ S1 S1 165 1 1 1 LOGIC___0]
[ S1 S1 40 1 1 1 LOGIC___1]
[ S1 S1 39 1 1 1 LOGIC___1]
[ S1 S1 166 1 2 2 LOGIC___0]
[ S1 S1 38 1 1 1 LOGIC___1]
[ S1 S1 37 1 1 1 LOGIC___1]
[ S1 S1 166 1 1 1 LOGIC___0]
[ S2 S1 36 1 1 1 LOGIC___0]
```

```
[ S1 S1 167 1 1 1 LOGIC__0]
[ S1 S1 35 1 1 1 LOGIC__1]
[ S1 S1 34 1 1 1 LOGIC__1]
[ S1 S1 167 1 2 2 LOGIC__0]
[ S1 S1 168 1 2 2 LOGIC__0]
[ S1 S1 32 1 1 1 LOGIC__1]
[ S1 S1 168 1 1 1 LOGIC__0]
[ S1 S1 31 1 1 1 LOGIC__1]
[ S1 S1 169 1 2 2 LOGIC__0]
[ S1 S1 30 1 1 1 LOGIC__1]
[ S1 S1 29 1 1 1 LOGIC__1]
[ S1 S1 169 1 1 1 LOGIC__0]
[ S1 S1 28 1 1 1 LOGIC__1]
[ S1 S1 170 1 2 2 LOGIC__0]
[ S1 S1 25 1 1 1 LOGIC__1]
[ S1 S1 170 1 1 1 LOGIC__0]
[ S1 S1 24 1 1 1 LOGIC__1]
[ S1 S1 171 1 1 1 LOGIC__0]
[ S2 S1 23 1 1 1 LOGIC__0]
[ S1 S1 22 1 1 1 LOGIC__1]
[ S2 S1 21 1 1 1 LOGIC__0]
[ S1 S1 172 1 1 1 LOGIC__0]
[ S1 S1 19 1 1 1 LOGIC__0]
[ S1 S1 173 1 1 1 LOGIC__0]
[ S1 S1 18 1 1 1 LOGIC__0]
[ S1 S1 19 1 2 2 LOGIC__0]
[ S1 S1 18 1 2 2 LOGIC__0]
[ S1 S1 17 1 1 1 LOGIC__0]
[ S1 S1 16 1 1 1 LOGIC__0]
[ S1 S1 17 1 2 2 LOGIC__0]
[ S1 S1 16 1 2 2 LOGIC__0]
[ S1 S1 174 1 1 1 LOGIC__0]
[ S1 S1 15 1 1 1 LOGIC__0]
[ S1 S1 14 1 1 1 LOGIC__0]
[ S1 S1 175 1 1 1 LOGIC__0]
[ S1 S1 13 1 1 1 LOGIC__0]
[ S1 S1 12 1 1 1 LOGIC__0]
[ S1 S1 11 1 1 1 LOGIC__0]
[ S1 S1 12 1 2 2 LOGIC__0]
[ S1 S1 11 1 2 2 LOGIC__0]
[ S1 S1 176 1 1 1 LOGIC__0]
[ S1 S1 10 1 1 1 LOGIC__0]
[ S1 S1 10 1 2 2 LOGIC__0]
[ S1 S1 9 1 1 1 LOGIC__0]
[ S1 S1 8 1 1 1 LOGIC__0]
[ S1 S1 177 1 1 1 LOGIC__0]
[ S1 S1 7 1 1 1 LOGIC__0]
[ S1 S1 7 1 2 2 LOGIC__0]
[ S1 S1 6 1 1 1 LOGIC__0]
[ S1 S1 6 1 2 2 LOGIC__0]
[ S1 S1 178 1 2 2 LOGIC__0]
[ S1 S1 5 1 1 1 LOGIC__0]
[ S1 S1 178 1 1 1 LOGIC__0]
[ S1 S1 4 1 1 1 LOGIC__0]
[ S1 S1 179 1 1 1 LOGIC__0]
[ S1 S1 2 1 1 1 LOGIC__0]
[ S1 S1 2 1 2 2 LOGIC__0]
[ S1 S1 1 1 1 1 LOGIC__0]
[ S1 S1 1 1 2 2 LOGIC__0]
[ S1 S1 180 1 1 1 LOGIC__0]
[ S1 S1 360 1 1 1 LOGIC__0]
[ S1 S1 359 1 1 1 LOGIC__0]
[ S1 S1 181 1 1 1 LOGIC__0]
[ S1 S1 357 1 1 1 LOGIC__0]
[ S1 S1 182 1 1 1 LOGIC__0]
[ S1 S1 355 1 1 1 LOGIC__0]
[ S1 S1 355 1 2 2 LOGIC__0]
[ S1 S1 354 1 1 1 LOGIC__0]
[ S1 S1 353 1 1 1 LOGIC__0]
[ S1 S1 353 1 2 2 LOGIC__0]
[ S1 S1 183 1 1 1 LOGIC__0]
[ S1 S1 352 1 1 1 LOGIC__0]
[ S1 S1 350 1 1 1 LOGIC__0]
[ S1 S1 349 1 2 2 LOGIC__0]
[ S1 S1 350 1 2 2 LOGIC__0]
[ S1 S1 184 1 1 1 LOGIC__0]
[ S1 S1 348 1 1 1 LOGIC__0]
[ S1 S1 185 1 1 1 LOGIC__0]
[ S1 S1 347 1 1 1 LOGIC__0]
[ S1 S1 346 1 1 1 LOGIC__0]
[ S1 S1 345 1 2 2 LOGIC__0]
[ S1 S1 346 1 2 2 LOGIC__0]
[ S1 S1 185 1 2 2 LOGIC__0]
[ S1 S1 345 1 1 1 LOGIC__0]
[ S1 S1 186 1 1 1 LOGIC__0]
[ S1 S1 344 1 1 1 LOGIC__0]
[ S1 S1 343 1 1 1 LOGIC__0]
[ S1 S1 187 1 1 1 LOGIC__0]
[ S1 S1 341 1 1 1 LOGIC__0]
[ S1 S1 340 1 1 1 LOGIC__2]
[ S1 S1 188 1 1 1 LOGIC__0]
[ S1 S1 338 1 1 1 LOGIC__2]
[ S1 S1 189 1 1 1 LOGIC__0]
[ S1 S1 335 1 1 1 LOGIC__2]
[ S1 S2 336 1 1 1 LOGIC__0]
[ S1 S1 325 1 1 1 LOGIC__2]
[ S1 S1 190 1 1 1 LOGIC__0]
[ S1 S1 190 1 2 2 LOGIC__0]
[ S1 S1 333 1 1 1 LOGIC__0]
[ S1 S1 332 1 2 2 LOGIC__0]
[ S1 S1 333 1 2 2 LOGIC__0]
[ S1 S1 332 1 1 1 LOGIC__2]
[ S1 S2 332 1 1 1 LOGIC__0]
[ S1 S1 331 1 1 1 LOGIC__2]
[ S1 S2 331 1 1 1 LOGIC__0]
[ S1 S1 241 1 1 1 LOGIC__2]
[ S1 S1 192 1 1 1 LOGIC__0]
[ S1 S1 192 1 2 2 LOGIC__0]
[ S1 S1 191 1 2 2 LOGIC__0]
[ S1 S1 329 1 1 1 LOGIC__2]
[ S1 S2 329 1 1 1 LOGIC__0]
[ S1 S1 191 1 1 1 LOGIC__0]
[ S1 S1 328 1 1 1 LOGIC__2]
[ S1 S2 328 1 1 1 LOGIC__0]
[ S1 S1 289 1 1 1 LOGIC__2]
[ S1 S1 327 1 1 1 LOGIC__2]
[ S1 S1 326 1 1 1 LOGIC__2]
[ S1 S1 193 1 1 1 LOGIC__0]
[ S1 S1 193 1 2 2 LOGIC__0]
[ S1 S1 194 1 1 1 LOGIC__0]
[ S1 S1 194 1 2 2 LOGIC__0]
[ S1 S1 323 1 1 1 LOGIC__2]
[ S1 S2 324 1 1 1 LOGIC__0]
[ S1 S1 284 1 1 1 LOGIC__2]
[ S1 S1 321 1 1 1 LOGIC__2]
[ S1 S2 322 1 1 1 LOGIC__0]
[ S1 S1 320 1 1 1 LOGIC__2]
[ S1 S2 321 1 1 1 LOGIC__0]
[ S1 S1 296 1 1 1 LOGIC__2]
[ S1 S1 195 1 1 1 LOGIC__0]
[ S1 S1 195 1 2 2 LOGIC__0]
[ S1 S1 319 1 1 1 LOGIC__0]
[ S1 S1 196 1 1 1 LOGIC__0]
[ S1 S1 196 1 2 2 LOGIC__0]
[ S1 S1 317 1 1 1 LOGIC__2]
[ S1 S1 316 1 1 1 LOGIC__2]
[ S1 S1 315 1 1 1 LOGIC__2]
[ S1 S1 197 1 1 1 LOGIC__0]
[ S1 S1 197 1 2 2 LOGIC__0]
[ S1 S1 314 1 1 1 LOGIC__2]
[ S1 S1 311 1 1 1 LOGIC__0]
[ S1 S1 198 1 1 1 LOGIC__0]
[ S1 S1 198 1 2 2 LOGIC__0]
[ S1 S1 310 1 1 1 LOGIC__2]
[ S1 S1 307 1 1 1 LOGIC__0]
[ S1 S1 199 1 1 1 LOGIC__0]
[ S1 S1 199 1 2 2 LOGIC__0]
[ S1 S1 306 1 1 1 LOGIC__2]
[ S1 S1 305 1 1 1 LOGIC__2]
[ S1 S1 304 1 1 1 LOGIC__2]
[ S1 S1 200 1 1 1 LOGIC__0]
[ S1 S1 200 1 2 2 LOGIC__0]
[ S1 S1 301 1 1 1 LOGIC__2]
[ S1 S2 302 1 1 1 LOGIC__0]
[ S1 S1 279 1 1 1 LOGIC__2]
[ S1 S1 300 1 1 1 LOGIC__2]
[ S1 S1 299 1 1 1 LOGIC__2]
[ S1 S1 201 1 1 1 LOGIC__2]
[ S1 S2 200 1 1 1 LOGIC__0]
[ S1 S2 297 1 1 1 LOGIC__0]
[ S1 S1 275 1 1 1 LOGIC__2]
[ S1 S1 294 1 1 1 LOGIC__2]
[ S1 S1 202 1 1 1 LOGIC__2]
[ S1 S2 201 1 1 1 LOGIC__0]
[ S1 S1 292 1 1 1 LOGIC__2]
[ S1 S5 202 1 1 1 LOGIC__0]
[ S1 S1 291 1 1 1 LOGIC__2]
[ S1 S1 203 1 1 1 LOGIC__2]
[ S1 S2 202 1 1 1 LOGIC__0]
[ S1 S1 288 1 1 1 LOGIC__2]
[ S1 S1 287 1 1 1 LOGIC__2]
[ S1 S2 203 1 1 1 LOGIC__0]
[ S1 S1 204 1 1 1 LOGIC__2]
[ S1 S1 282 1 1 1 LOGIC__2]
[ S1 S1 280 1 1 1 LOGIC__2]
[ S1 S2 204 1 1 1 LOGIC__0]
[ S1 S1 205 1 1 1 LOGIC__0]
[ S1 S1 205 1 2 2 LOGIC__0]
[ S1 S1 204 1 2 2 LOGIC__0]
[ S1 S1 277 1 1 1 LOGIC__2]
[ S1 S1 274 1 1 1 LOGIC__2]
[ S1 S1 206 1 1 1 LOGIC__2]
[ S1 S2 205 1 1 1 LOGIC__0]
[ S1 S1 265 1 1 1 LOGIC__2]
[ S1 S1 207 1 1 1 LOGIC__2]
[ S1 S2 206 1 1 1 LOGIC__0]
[ S1 S1 260 1 1 1 LOGIC__2]
[ S1 S1 256 1 1 1 LOGIC__2]
[ S1 S1 207 1 2 2 LOGIC__0]
[ S1 S1 206 1 2 2 LOGIC__0]
[ S1 S1 208 1 1 1 LOGIC__2]
[ S1 S2 207 1 1 1 LOGIC__0]
```

```
[ S1 S1 209  1  1  1  LOGIC___2]
[ S1 S2 208  1  1  1  LOGIC___0]
[ S1 S1 210  1  1  1  LOGIC___2]
[ S1 S2 209  1  1  1  LOGIC___0]
[ S1 S1 211  1  1  1  LOGIC___2]
[ S1 S2 210  1  1  1  LOGIC___0]
[ S1 S1 213  1  1  1  LOGIC___2]
[ S1 S2 211  1  1  1  LOGIC___0]
[ S1 S1 212  1  1  1  LOGIC___2]
[ S1 S2 212  1  1  1  LOGIC___0]
[ S1 S1 213  1  2  2  LOGIC___0]
[ S1 S1 212  1  2  2  LOGIC___0]
[ S1 S1 214  1  1  1  LOGIC___2]
[ S1 S2 213  1  1  1  LOGIC___0]
[ S1 S1 215  1  1  1  LOGIC___2]
[ S1 S2 214  1  1  1  LOGIC___0]
[ S1 S1 215  1  2  2  LOGIC___0]
[ S1 S1 214  1  2  2  LOGIC___0]
[ S1 S2 215  1  1  1  LOGIC___0]
[ S1 S1 216  1  1  1  LOGIC___2]
[ S1 S1 210  1  2  2  LOGIC___0]
[ S1 S1 209  1  2  2  LOGIC___0]
[ S1 S1 201  1  2  2  LOGIC___0]
[ S1 S1 189  1  2  2  LOGIC___0]
[ S1 S1 188  1  2  2  LOGIC___0]
[ S1 S1 187  1  2  2  LOGIC___0]
[ S1 S1 186  1  2  2  LOGIC___0]
[ S1 S1 184  1  2  2  LOGIC___0]
[ S1 S1 183  1  2  2  LOGIC___0]
[ S1 S1 182  1  2  2  LOGIC___0]
[ S1 S1 181  1  2  2  LOGIC___0]
[ S1 S1 180  1  2  2  LOGIC___0]
[ S1 S1 179  1  2  2  LOGIC___0]
[ S1 S1 177  1  2  2  LOGIC___0]
[ S1 S1 176  1  2  2  LOGIC___0]
[ S1 S1 175  1  2  2  LOGIC___0]
[ S1 S1 174  1  2  2  LOGIC___0]
[ S1 S1 173  1  2  2  LOGIC___0]
[ S1 S1 171  1  2  2  LOGIC___0]
[ S1 S1 160  1  2  2  LOGIC___0]
[ S2 S1 158  1  1  1  LOGIC___0]
[ S2 S1 157  1  1  1  LOGIC___0]
] -> Example_G40;
id3(Example_G40,Attribute_G40,C
lass_G40)->Rule_G40;
print_rule(Rule_G40);
pf(Rule_G40,6);nl(1);
vars Pruning Level;
true -> Pruning;
99 -> Level;
id3_chisqd(Example_G40,Attribute
_G40,Class_G40,
        Level,Pruning) ->Rule_G40;
print_rule(Rule_G40);
pfx(Rule_G40,6);nl(1);
```

APPENDIX E

CONFLICT DATA IN LEARNIG SAMPLE

```
;;;________________Conflict
Data in 5in Guns Learning Sam-
ple________________
[ S1 S1 144 2 0 0 LOGIC___0]
[ S1 S1 141 2 0 0 LOGIC___0]
[ S1 S1 138 2 0 0 LOGIC___0]
[ S1 S1 142 1 0 0 LOGIC___0]
[ S1 S1 112 2 0 0 LOGIC___0]
[ S1 S1 143 1 0 0 LOGIC___0]
[ S1 S1 141 1 0 0 LOGIC___0]
[ S1 S1 44 2 0 0 LOGIC___0]
[ S1 S1 185 1 1 1 LOGIC___0]
[ S1 S1 213 1 0 0 LOGIC___0]
[ S1 S1 219 1 0 0 LOGIC___0]
[ S1 S1 218 2 0 0 LOGIC___0]
[ S1 S1 215 2 0 0 LOGIC___0]
[ S1 S1 213 2 0 0 LOGIC___0]
[ S1 S1 212 2 0 0 LOGIC___0]
[ S1 S1 211 2 0 0 LOGIC___0]
[ S1 S1 209 2 0 0 LOGIC___0]
[ S1 S1 208 2 0 0 LOGIC___0]
[ S1 S1 206 2 0 0 LOGIC___0]
[ S1 S1 204 2 0 0 LOGIC___0]
[ S1 S1 203 2 0 0 LOGIC___0]
[ S1 S1 201 2 0 0 LOGIC___0]
[ S1 S1 200 2 0 0 LOGIC___0]
[ S1 S1 199 2 0 0 LOGIC___0]
[ S1 S1 190 2 0 0 LOGIC___0]
[ S1 S1 187 2 0 0 LOGIC___0]
[ S1 S1 183 2 0 0 LOGIC___0]
[ S1 S1 181 2 0 0 LOGIC___0]
[ S1 S1 178 2 0 0 LOGIC___0]
[ S1 S1 173 2 0 0 LOGIC___0]
[ S1 S1 163 2 0 0 LOGIC___0]
[ S1 S1 156 2 0 0 LOGIC___0]
[ S1 S1 155 2 0 0 LOGIC___0]
[ S1 S1 152 2 0 0 LOGIC___0]
;;;****** TOTAL REMOTE = 34
******;;;
```

```
;;;________________Conflict
Data in 76mm Guns Learning Sam-
ple________________
[ S1 105 1 0 LOGIC___0]
[ S1 98 1 0 LOGIC___0]
[ S1 76 1 0 LOGIC___0]
[ S1 75 1 0 LOGIC___0]
[ S1 72 1 0 LOGIC___0]
[ S1 71 1 0 LOGIC___0]
[ S1 42 1 0 LOGIC___0]
[ S1 37 1 0 LOGIC___0]
[ S1 28 1 0 LOGIC___0]
[ S1 255 1 0 LOGIC___0]
;;;****** TOTAL REMOTE = 10
******;;;
```

```
;;;________________Conflict
Data in 40mm Guns Learning Sam-
ple________________
[ S1 S1 333 1 1 1 LOGIC___0]
[ S1 S1 319 1 1 1 LOGIC___0]
[ S1 S1 311 1 1 1 LOGIC___0]
[ S1 S1 307 1 1 1 LOGIC___0]
[ S1 S1 200 1 1 1 LOGIC___0]
[ S1 S1 205 1 1 1 LOGIC___0]
;;;****** TOTAL REMOTE = 6
******;;;
```

APPENDIX F

INDUCTION RULES OF WEAPON ASSIGNMENT GENERATED BY ID3 AND χ^2 ALGORITHM

```
=========== 5in Guns Assignment Induction Rules Generated by ID3 Algorithm ============
SURG5N(1,1)
S1 : SURG5N(1,2)
 S1 : ANGLE
  < 220 : ANGLE
      < 140 : LOGIC=1
      >= 140 : PRIOR(NUMTGT)
          < 3 : LOGIC=2
          >= 3 : LOGIC=0
  >= 220 : LOGIC=1
 S2 : LOGIC=0
 S5 : NULL
S2 : PRIOR(TGTG5N(1))
 < 2 : ANGLE
    < 180 : ANGLE
        < 51 : LOGIC=0
        >= 51 : ANGLE
            < 53 : LOGIC=2
            >= 53 : ANGLE
                < 54 : LOGIC=0
                >= 54 : LOGIC=2
    >= 180 : LOGIC=0
 >= 2 : ANGLE
     < 289 : ANGLE
       < 277 : ANGLE
         < 276 : ANGLE
           < 219 : ANGLE
             < 136 : ANGLE
               < 135 : ANGLE
                 < 131 : ANGLE
                   < 128 : ANGLE
                     < 121 : ANGLE
                       < 120 : ANGLE
                         < 98 : ANGLE
                           < 97 : ANGLE
                             < 93 : ANGLE
                               < 92 : ANGLE
                                 < 88 : ANGLE
                                   < 87 : ANGLE
                                     < 80 : ANGLE
                                       < 79 : ANGLE
                                        < 75 : ANGLE
                                          < 74 : ANGLE
                                            < 64 : ANGLE
                                              < 57 : ANGLE
                                                < 56 : LOGIC=2
                                                >= 56 : LOGIC=0
                                              >= 57 : LOGIC=2
                                            >= 64 : ANGLE
                                               < 65 : LOGIC=0
                                               >= 65 : ANGLE
                                                  < 67 : ANGLE
                                                    < 66 : LOGIC=2
                                                    >= 66 : LOGIC=0
```

```
                              >= 67 : ANGLE
                                <  72 : LOGIC=2
                                >= 72 : ANGLE
                                   < 73 : LOGIC=0
                                   >= 73 : LOGIC=2
                          >= 74 : LOGIC=0
                         >= 75 : LOGIC=2
                       >= 79 : LOGIC=0
                     >= 80 : LOGIC=2
                   >= 87 : LOGIC=0
                  >= 88 : LOGIC=2
                >= 92 : LOGIC=0
              >= 93 : LOGIC=2
            >= 97 : LOGIC=0
          >= 98 : ANGLE
             < 108 : LOGIC=2
             >= 108 : ANGLE
               < 109 : LOGIC=0
               >= 109 : ANGLE
                 < 116 : LOGIC=2
                 >= 116 : ANGLE
                   < 117 : LOGIC=0
                   >= 117 : LOGIC=2
                     >= 120 : LOGIC=0
                        >= 121: LOGIC=2
                          >= 128 : ANGLE
                             <129 : LOGIC=0
                             >= 129 : ANGLE
                               < 130 : LOGIC=2
                               >= 130 : LOGIC=0
   >= 131 : LOGIC=2
  >= 135 : LOGIC=0
 >= 136 : LOGIC=2
>= 219 : ANGLE
< 224 : ANGLE
    < 221 : ANGLE
       < 220 : LOGIC=0
       >= 220 : LOGIC=2
    >= 221 : LOGIC=0
>= 224 : ANGLE
    < 259 : ANGLE
        < 251 : ANGLE
           < 250 : ANGLE
               < 246: ANGLE
                 < 245 : ANGLE
                  < 240 : ANGLE
                    < 239 : ANGLE
                      < 234 : ANGLE
                        < 233 : ANGLE
                          < 227 : LOGIC=2
                          >= 227 : ANGLE
                             < 229 : LOGIC=0
                             >= 229 : LOGIC=2
                        >= 233 : LOGIC=0
                       >= 234 : LOGIC=2
                     >= 239 : LOGIC=0
                    >= 240 : LOGIC=2
                >= 245 : LOGIC=0
               >= 246  : LOGIC=2
           >= 250 : LOGIC=0
        >= 251 : LOGIC=2
    >= 259 : ANGLE
        < 273 : ANGLE
           < 260 : LOGIC=0
```

```
                                   >= 260 : ANGLE
                                     < 272  : ANGLE
                                       < 262 : LOGIC=2
                                       >= 262 : ANGLE
                                         < 263 : LOGIC=0
                                         >= 263 : ANGLE
                                           < 269 : ANGLE
                                             < 268 : ANGLE
                                               < 265 : ANGLE
                                                 < 264 : LOGIC=2
                                                 >= 264 : LOGIC=0
                                               >= 265 : LOGIC=2
                                             >= 268 : LOGIC=0
                                           >= 269 : LOGIC=2
                                       >=  272 : LOGIC=0
                                 >= 273 : LOGIC=2
          >= 276 : LOGIC=0
       >= 277 : LOGIC=2
    >= 289 : LOGIC=0
S5 : NULL
============ 76mm Gun Assignment Induction Rules Generated by ID3 Algorithm =============


SURG76(1,1)
S0 : NULL
S1 : ANGLE
 < 255 : ANGLE
     < 104 : LOGIC=1
     >= 104 : LOGIC=0
 >= 255 : LOGIC=1
S2 : LOGIC=0
S5 : NULL

============ 40mm Guns Assignment Induction Rules Generated by ID3 Algorithm =============

ANGLE
< 161 : ANGLE
    < 22 : LOGIC=0
    >= 22 : SURG40(1,1)
        S1 : PRIOR(TGTG40(1))
         < 2 : LOGIC=1
         >= 2 : LOGIC=0
        S2 : LOGIC=0
        S5 : LOGIC=0
>= 161 : ANGLE
    < 201 : LOGIC=0
    >= 201 : SURG40(1,2)
         S1 : ANGLE
          < 341 : PRIOR(TGTG40(1))
              < 2 : LOGIC=2
              >= 2 : LOGIC=0
          >= 341 : LOGIC=0
         S2 : LOGIC=0
         S5 : LOGIC=0
============= 5in Guns Assignment  Induction Rules Generated by Chi_squared Algorithm =======
PRIOR(TGTG5N(1))
< 1 : SURG5N(1,2)
   S1 : PRIOR(NUMTGT)
    < 3 : ANGLE
       < 220 : ANGLE
           < 140 : LOGIC=1 197
           >= 140 : LOGIC=2 121
       >= 220 : LOGIC=1 226
    >= 3 : LOGIC=0 1
```

```
   S2 : LOGIC=0 75
   S5 : NULL
>= 1 : ANGLE
   < 289 : ANGLE
       < 51 : SURG5N(1,1)
           S1 : LOGIC=1 8
           S2 : LOGIC=0 50
           S5 : NULL
       >= 51 : LOGIC=0 31
            LOGIC=2 169
            LOGIC=1 0
   >= 289 : SURG5N(1,1)
        S1 : LOGIC=1 3
        S2 : LOGIC=0 73
        S5 : NULL
```

=========== 76mm Gun Assignment Induction Rules Generated by Chi_squared Algorithm ========

```
SURG76(1,1)
S0 : NULL
S1 : ANGLE
 < 255 : ANGLE
      < 104 : LOGIC=1 101
      >= 104 : LOGIC=0 128
 >= 255 : LOGIC=1 108
S2 : LOGIC=0 193
S5 : NULL
```

=========== 40mm Guns Assignment Induction Rules Generated by Chi_squared Algorithm ========

```
ANGLE
< 161 : ANGLE
    < 22 : LOGIC=0 30
    >= 22 : SURG40(1,1)
         S1 : PRIOR(TGTG40(1))
          < 2 : LOGIC=1 65
          >= 2 : LOGIC=0 3
         S2 : LOGIC=0 25
         S5 : LOGIC=0 2
>= 161 : ANGLE
     < 201 : LOGIC=0 80
     >= 201 : SURG40(1,2)
          S1 : ANGLE
           < 341 : PRIOR(TGTG40(1))
                < 2 : LOGIC=2 57
                >= 2 : LOGIC=0 13
           >= 341 : LOGIC=0 21
          S2 : LOGIC=0 25
          S5 : LOGIC=0 1
```

APPENDIX G

TRANSFORMED FORTRAN CODE OF REFINED INDUCTION RULES FOR WEAPON ASSIGNMENT

```
C_____________________Rule for 5in Guns Allocation____________________________
C
      IF (PRIOR(TGTG5N(1)) .LT. 1) THEN
       IF ( SURG5N(1,2) .EQ. 1) THEN
         IF (PRIOR(NUMTGT) .LT. 3) THEN
          IF (ANGLE .LT. 220) THEN
             IF (ANGLE .LT. 140) THEN
                 LOGIC=1
                 RETURN
             ELSE
                 LOGIC=2
                 RETURN
             ENDIF
          ELSE
              LOGIC=1
              RETURN
          ENDIF
         ELSE
           LOGIC=0
           RETURN
         ENDIF
       ENDIF
       IF ( SURG5N(1,2) .EQ. 2) THEN
          LOGIC=0
          RETURN
       ENDIF
       IF ( SURG5N(1,2) .EQ. 5) THEN
          RETURN
       ENDIF
      ELSE
       IF (ANGLE .LT. 289) THEN
          IF (ANGLE .LT. 51) THEN
            IF ( SURG5N(1,1) .EQ. 1) THEN
               LOGIC=1
               RETURN
            ENDIF
            IF ( SURG5N(1,1) .EQ. 2) THEN
               LOGIC=0
               RETURN
            ENDIF
            IF ( SURG5N(1,1) .EQ. 5) THEN
               RETURN
            ENDIF
          ELSE
             LOGIC=2
             RETURN
          ENDIF
       ELSE
          IF ( SURG5N(1,1) .EQ. 1) THEN
             LOGIC=1
             RETURN
          ENDIF
          IF ( SURG5N(1,1) .EQ. 2) THEN
```

```
              LOGIC=0
              RETURN
           ENDIF
           IF ( SURG5N(1,1) .EQ. 5) THEN
              RETURN
           ENDIF
        ENDIF
      ENDIF
C
C____________________Rule for 76mm Gun Allocation___________________________
C
      IF ( SURG76(1,1) .EQ. 0) THEN
         RETURN
      ENDIF
      IF ( SURG76(1,1) .EQ. 1) THEN
        IF (ANGLE .LT. 255) THEN
           IF (ANGLE .LT. 104) THEN
               LOGIC=1
               RETURN
           ELSE
               LOGIC=0
               RETURN
           ENDIF
        ELSE
            LOGIC=1
            RETURN
        ENDIF
      ENDIF
      IF ( SURG76(1,1) .EQ. 2) THEN
         LOGIC=0
         RETURN
      ENDIF
      IF ( SURG76(1,1) .EQ. 5) THEN
         RETURN
      ENDIF
C
C____________________Rule for 40mm Guns Allocation__________________________
C
      IF (ANGLE .LT. 161) THEN
         IF (ANGLE .LT. 22) THEN
            LOGIC=0
            RETURN
         ELSE
           IF ( SURG40(1,1) .EQ. 1) THEN
             IF (PRIOR(TGTG40(1)) .LT. 2) THEN
               LOGIC=1
               RETURN
             ELSE
               LOGIC=0
               RETURN
             ENDIF
           ENDIF
           IF ( SURG40(1,1) .EQ. 2) THEN
              LOGIC=0
              RETURN
           ENDIF
           IF ( SURG40(1,1) .EQ. 5) THEN
              LOGIC=0
              RETURN
           ENDIF
         ENDIF
      ELSE
         IF (ANGLE .LT. 201) THEN
             LOGIC=0
```

```
        RETURN
    ELSE
       IF ( SURG40(1,2) .EQ. 1) THEN
         IF (ANGLE .LT. 341) THEN
            IF (PRIOR(TGTG40(1)) .LT. 2) THEN
              LOGIC=2
              RETURN
            ELSE
              LOGIC=0
              RETURN
            ENDIF
         ELSE
             LOGIC=0
             RETURN
         ENDIF
       ENDIF
       IF ( SURG40(1,2) .EQ. 2) THEN
          LOGIC=0
          RETURN
       ENDIF
       IF ( SURG40(1,2) .EQ. 5) THEN
          LOGIC=0
          RETURN
       ENDIF
    ENDIF
 ENDIF
```

APPENDIX H

SOURCE PROGRAM FOR TRANSLATING DECISION TREE INTO FORTRAN CODE

```
;;;   ID3_rule ==> FORTRAN Code Generator.
;;;_____________________________________________________
define cr(indent);
lvars indent;
   nl(1);sp(indent);
enddefine;
;;;_____________________________________________________
define pr_cnti(bnd) -> cnt;
   length([%dest_characters(bnd)%]) -> cnt;
   if bnd = "<" then
      pr(' .LT. ');
   else pr(bnd);
   endif;
enddefine;
;;;_____________________________________________________
define pr_cnt(bnd) -> cnt;
   length([%dest_characters(bnd)%]) -> cnt;
   pr('\'');pr(bnd);pr('\'');
enddefine;
;;;_____________________________________________________
define pf(rule,indent);
   vars rule indent att lb lower ub upper cname cnt;
   if rule matches [INTEGER ?att [?lb ?lower] [?ub ?upper]] then
         cr(indent);pr('IF (');pr(att);pr_cnti(hd(lb))->cnt;
         pr_cnti(hd(tl(lb))) -> cnt;pr(') THEN');
         pf(lower,indent+cnt);
         cr(indent);pr('ELSE      ');
         pf(upper,indent+cnt);
         cr(indent);pr('ENDIF');
   elseif rule matches [LOGICAL ==] then
                       pplogical(rule,indent);
   elseif rule matches [CLASS  = ?cname] then
                       cr(indent);pr(' ');pr(cname);
                       cr(indent);pr(' RETURN');
   elseif rule matches [NULL] then
                       cr(indent);pr(' RETURN');
   endif;
enddefine;
;;;_____________________________________________________
define pplogical(rule,indent);
vars rule indent att valuelist value val subtree;
   rule --> [LOGICAL ?att ??valuelist];
   for value in valuelist do
       value --> [?val ?subtree];
       cr(indent);spr('IF (');spr(att);spr('.EQ.');pr_cnt(val)->cnt;
       spr(') THEN');
       pf(subtree,indent+cnt);
       cr(indent);pr('ENDIF');
   endfor;
enddefine;
;;;_____________________________________________________
;;; pick up duplicated item after sort
define pickup(list);
```

```
until length(list) < 2 do
   if hd(list) matches hd(tl(list))
      then     tl(list) -> list;
      else     add(hd(list));
               tl(list) -> list;
   endif;
enduntil;
add(hd(list));
enddefine;
;;;_____________________________________________
define pfx(rule,indent);
   vars rule indent att lb lower ub upper cname cnt freq;
   if rule matches [INTEGER ?att [?lb ?lower] [?ub ?upper]] then
         cr(indent);pr('IF (');pr(att);pr_cnti(hd(lb))->cnt;
         pr_cnti(hd(tl(lb))) -> cnt;pr(') THEN');
         pfx(lower,indent+cnt);
         cr(indent);pr('ELSE      ');
         pfx(upper,indent+cnt);
         cr(indent);pr('ENDIF');
   elseif rule matches [LOGICAL ==] then
                       pplogicalx(rule,indent);
   elseif rule matches [CLASS = ?cname ?freq ==] then
                       cr(indent);pr(' ');pr(cname);
                       cr(indent);pr(' RETURN');
   elseif rule matches [[PROB_CLASS ==] ==] then
                       pr_prob_class(rule,indent);
   elseif rule matches [NULL] then
                       cr(indent);pr(' RETURN');
   endif;
enddefine;
;;;_____________________________________________
define pr_prob_class(rule,indent);

lvars rule indent node;
vars class freq;

   for node in rule do
      node --> [PROB_CLASS = ?class ?freq ==];
      cr(indent);pr(' ');pr(class);pr(' ');pr(freq);
      cr(indent);pr(' RETURN');
      if node/==last(rule) then cr(indent+1); endif;
   endfor;
enddefine;
;;;_____________________________________________
define pplogicalx(rule,indent);
vars rule indent att valuelist value val subtree;
   rule --> [LOGICAL ?att ??valuelist];
   for value in valuelist do
        value --> [?val ?subtree];
        cr(indent);spr('IF (');spr(att);spr('.EQ.');pr_cnt(val)->cnt;
        spr(') THEN');
        pfx(subtree,indent+cnt);
        cr(indent);pr('ENDIF');
   endfor;
enddefine;
;;;_____________________________________________
```

LIST OF REFERENCES

1. Anna Hart, "*Induction and Knowledge Elicitation*", pp. 165-188,
2. Jerry Banks, John S. Catson II, "*Discrete-Event System Simulation*", Prentice Hall, 1984.
3. The Office of the Secretary of Defense, "*Operational Suitability Guide*", Vol. I, 1990.
4. Beriman, Friedman, Olshen and Stone, "*Classification and Regression Tree*", Wadsworth 1984.
5. Buchanan, Sullivan, Cheng and Clearwater, "*Simulation-Assisted Inductive Learning*", Knowledge Systems Laboratory, Stanfor University, Stanford, California,
6. J. Ross Quinlan, "*Application of Expert System*", Addison-Wesley, 1987.
7. J. Ross Quinlan, "*Learning from Noisy Data*", Paper presented at the International Machine Learning Workshop, University of Illinois at Urbana Champaign, 1983.
8. J. Ross Quinlan, "*Probabilitic Decision Trees*", University of Sydney,
9. Mike Burton, Nigel Shadbolt, "*POP-11 Programming for Artificial Intelligence*", Addison-Wesley, 1987.
10. Jonathan Laventhol, "*Programming in POP-11*", Blackwell Scientific Publication, 1987.
11. John Mingers, "*Expert Systems - Experiments with Rule Induction*", J. Opl Res. Soc. Vol. 37, No. 11, pp. 1031-1037, 1986.
12. John Mingers, "*Expert Systems - Rule Induction with Statistical Data*", J. Opl Res. Soc. Vol. 38, No. 11, pp. 39-47, 1987.
13. John Mingers, "*Rule Induction with Statistical Data - A Comparison with Multiple Regression*", J. Opl Res. Soc. Vol. 38, No. 4, pp. 347-351, 1987.
14. Philip R. Race and Richard C. Thomas, "*Rule Induction in Investment Appraisal*", J. Opl Res. So. Vol. 39, No. 12, pp. 1113-1123, 1988.
15. Philip R. Race and Richard C. Thomas, "*Rule Induction Software*" Technical Report, School of Computer Studies, University of Leeds, England, 1988.
16. Zarchan Paul, "*Tactical and Strategic Missile Guidance*", American Institute of Aeronautics and Astronautics, Inc., Washington, D.C., 1990

17. Frank T. Heilenday, "*Principles of air Defense and Air Vehicle Penetration*", CEEPrees Books, Continuing Engineering Education George Washington University, Washington, D.C., 20052, 1988.

INITIAL DISTRIBUTION LIST

		No. of Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Dudley Knox Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Department Chairman, Code OR Department of Operations Research Naval Postgraduate School Monterey, CA 93943-5000	1
4	Department Chairman, Code CS Department of Computer Science Naval Postgraduate School Monterey, CA 93943	1
5.	Professor Yuh-jeng Lee, Code CS Department of Computer Science Naval Postgraduate School Monterey, CA 93943	5
6.	Professor Lyn R. Whitaker, Code OR Department of Operations Research Naval Postgraduate School Monterey, CA 93943-5000	1
7.	CDR Chang-yun Lo 3rd Fl, 12, Lane 34, Mo-Fan St. Taichung, Taiwan, R.O.C.	3